VISA POUR L'INFORMATIQUE

Jean-Michel Jégo



P.S.I.

VISA POUR L'INFORMATIQUE

VOTRE BIBLIOTHEQUE D'INFORMATIQUE INDIVIDUELLE

Programmer en Assembleur — Alain Pinaud
Programmer en Basic — Michel Plouin
Le Basic et ses fichiers — Jacques Boisgontier
Programmer en L.S.E. — Stéphane Berche et Yves Noyelle
Programmer en Pascal — Daniel-Jean David et Jean-Luc Deschamps
Comment programmer Jean-Claude Barbance
Comprendre les microprocesseurs — Roland Dubois

La découverte de l'Applesoft - Frédéric Lévy et Dominique Schraen
La pratique de l'Apple II — Nicole Bréaud-Pouliquen

La pratique du LX500 — Alain Séméteys et Francis Vasse

La pratique du MZ-80K — Jean-Pierre Lhoir
La découverte du P.E.T./C.B.M. — Daniel-Jean David
La pratique du P.E.T./C.B.M. — Daniel-Jean David

La pratique du TRS-80 — Pierre Giraud et Alain Pinaud

ISBN : 2-86595-021-2

Imprimé en France

VISA POUR L'INFORMATIQUE

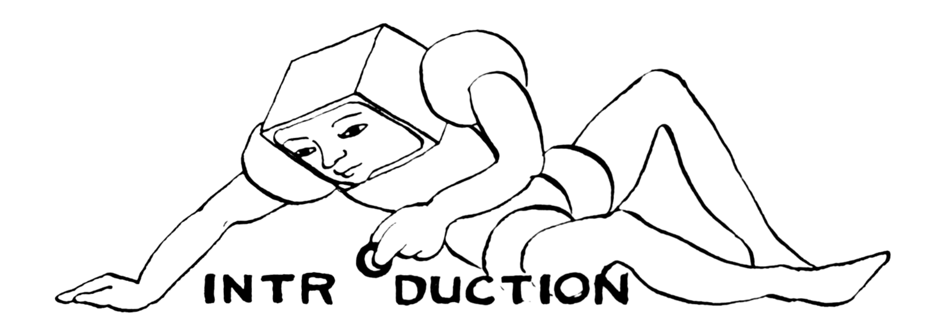
Jean-Michel Jégo



illustrations Françoise Guillot

Editions du P.S.I. – 1981

SOMMAIRE



«Vous avez dit Ordinateur ?» Combien de fois nous sommes-nous déjà trouvés face à cette interrogation ?

Dire que l'ordinateur est en passe de bouleverser toutes nos habitudes peut paraître un lieu commun. C'est en effet ce dont chaque jour, les médias se font l'écho. Nous voyons, à l'occasion d'applications souvent spectaculaires cet outil pénétrer les domaines les plus divers de la connaissance.

Ceux qui travaillent dans les entreprises ont vécu la mise en place de certaines applications informatisées : bulletin de paie, facturation de clients, gestion de la production ...

Peut-être le phénomène est-il encore plus perceptible dans notre environnement. Nous avons pu suivre sur des écrans de télévision, grâce à des vues de la salle des ordinateurs située à Houston, le suspense des premiers pas de l'homme sur la lune. C'est par l'utilisation simultanée de plusieurs ordinateurs que cet exploit de la technique astronautique fut possible.

Plus près de nous, dans la vie quotidienne, des systèmes informatiques plus ou moins sophistiqués permettent de tenir à jour nos comptes bancaires et postaux, interviennent dans le domaine de la médecine, des voyages, pour la réservation des hôtels ou des places d'avion ou de train, de l'architecture, pour le tracé de ponts ou d'autoroutes, sans oublier celui du commerce pour les encaissements et la tenue des stocks des grands magasins.

Et voici que maintenant, peu à peu, nous voyons l'informatique faire son entrée dans la vie domestique. Le parc des «petits systèmes individuels» ou (P.S.I.) se développe rapidement.

Outre la résolution de calculs mathématiques ou la réalisation de travaux de gestion, ces systèmes peuvent être utilisés à des fins ludiques, qu'il s'agisse de jouer aux échecs ou à la bataille navale, éventuellement musicales, ainsi qu'à la mémorisation de vocabulaire. Bientôt les traditionnels annuaires téléphoniques seront supplantés par la mise en place de renseignements informatisés.

Plus récemment encore, l'«ordinateur de poche» sait tenir à jour un fichier d'adresses, un carnet de rendez-vous, et effectue toutes sortes de calculs, du montant des intérêts à d'autres plus complexes.

Les progressistes optimistes prédisent la multiplication des domaines d'application, voire jusqu'à la programmation de nos menus quotidiens, tandis que les alarmistes attribuent à ces machines des facultés dévastatrices insoupçonnées.

Quels que soient les prolongements de ces systèmes dans les années à venir, chacun de nous, c'est certain, y sera impliqué. Aussi n'est-il pas tentant de familiariser les utilisateurs c'est-à-dire vous et moi, avec ces machines ? C'est ce à quoi tend cet ouvrage : permettre à toute personne, non spécialisée en informatique, de faire ses premiers pas dans un univers en pleine expansion.

* *
*

La lecture et l'utilisation de ce livre ne nécessitent à aucun moment de posséder chez soi un ordinateur. Les programmes sont écrits en français, sous-titrés en Anglais BASIC, car la plupart des matériels fonctionnent dans ce langage.

Possédez-vous un ordinateur ? Vous pourrez essayer les exercices qui vous sont proposés. Ils ont été choisis pour leur simplicité et peuvent être réalisés sur tous les types de matériels.

L'INFORMATIQUE OU LE TRAITEMENT DE L'INFORMATION

Qu'est-ce que l'informatique ? Le petit Larousse nous enseigne que l'informatique est la *«technique du traitement automatique de l'information»*.



Une telle définition mérite quelques éclaircissements. Il s'agit d'une technique qui s'appuie sur des lois et des régles. Cette discipline a ses propres contraintes et fait appel à des procédés et des méthodes particulièrement adaptés ; il n'est pas indispensable, pour utiliser une technique, d'en connaître tous les rouages.

De même qu'il n'est pas nécessaire de connaître la technique automobile pour conduire, il n'est pas indispensable d'étudier la technique informatique pour utiliser un ordinateur.

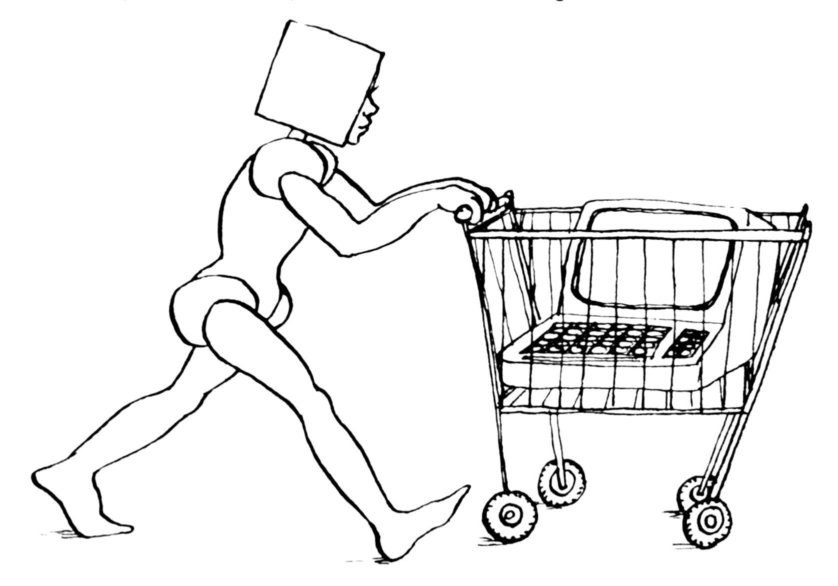
Quotidiennement, nous recevons quantités d'informations. Quelques-unes de celles-ci peuvent être traitées automatiquement, c'est-à-dire sans intervention directe de l'homme, et relèvent en ce sens de l'informatique. Examinons quelques exemples d'informations pris dans la vie courante.

Lorsque nous achetons une livre de beurre, nous recevons une information qui est le prix affiché au kilo. Pour connaître le montant à payer, il faut effectuer une opération. Cette opération fait appel à la connaissance antérieure du rapport entre livre et kilo, qui est nécessaire au traitement de l'information.

Nous plaçant une nouvelle fois dans cette situation d'achat, dans un magasin à grande surface, nous recevons des informations relatives à des prix, à des poids, à des volumes. Le traitement rapide des informations reçues nous permet de comparer les prix rapportés, soit au kilo, soit à toute autre unité de compte (paquet, douzaine ...). Bien entendu, à ces grandeurs mesurables et calculables, nous pouvons

ajouter des critères de sélection : couleur, efficacité du produit, goût, emballage ...

Notre choix se porte sur le produit qui a reçu le meilleur classement en fonction de l'ensemble des critères de sélection. Le traitement de l'information effectué consiste en un calcul, suivi de comparaisons pour terminer par un classement en fonction de nos critères ; notre décision d'achat peut d'ailleurs être tout à fait irrationnelle, et ne pas tenir compte du classement soigneusement établi.



NOTRE CHOIX SE PORTE SUR LE PRODUIT QUI A REÇU LE MEILLEUR CLASSEMENT...

Lorsque nous recherchons la définition d'un mot dans un dictionnaire, nous savons que les informations sont regroupées par ordre alphabétique, il nous est donc facile d'accéder tout d'abord à la lettre puis au mot lui-même.

Dans ce cas, le traitement de l'information consiste en une recherche à partir d'un classement alphabétique.

Ces trois exemples montrent que le traitement de l'information revêt les aspects suivants :

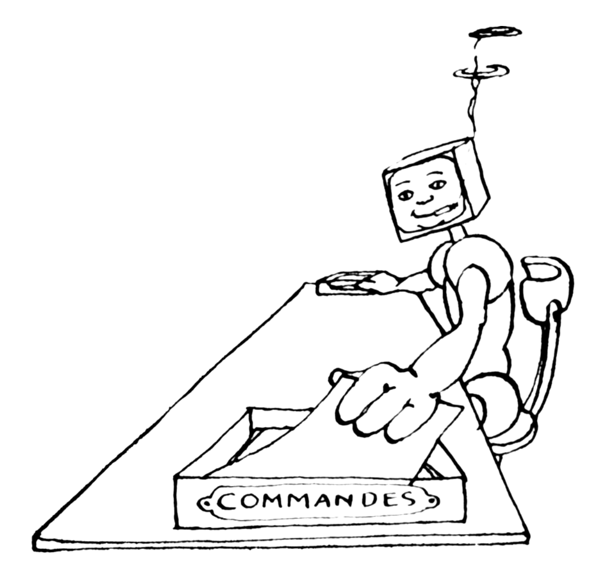
calcul
comparaison
tri
classement

La banalité de ces situations nous fait percevoir que nous baignons dans un flot d'informations, qui doivent être traitées pour devenir utilisables. Certaines d'entre-elles peuvent l'être de façon automatique grâce à des machines dont l'ordinateur est la dernière née.

Le second chapitre de ce livre nous présentera ce qu'est l'ordinateur même

Pour l'instant, essayons de déterminer par un nouvel exemple ce qui, dans une démarche de traitement de l'information peut être automatisé ou non.

Envisageons le cas traditionnel d'un employé assis à sa table de travail et qui appartiendrait à une société de vente par correspondance. Il reçoit des commandes et émet des bons de livraison accompagnés de factures. Notre employé consciencieux est en plus méthodique : il place les commandes à gauche et les factures à droite.



NOTRE EMPLOYÉ CONSCIENCIEUX
EST, EN PLUS, METHODIQUE

Comment travaille-t-il ?

Il prend une *commande* et il la lit. Celle-ci comprend :

- le nom du client
- éventuellement un numéro de compte client
- la liste et la quantité de produits commandés
- le lieu de la livraison

Pour effectuer son travail, notre employé va consulter un *fichier client.* Il en sort une fiche sur laquelle sont reportés :

- les commandes antérieures
- l'état du compte de ce client
  est-il à jour ? a-t-il des dettes ?
- d'autres informations commerciales éventuellement telles que le taux de remise par rapport aux prix du catalogue.

Si le client n'a pas de fiche, il devra en créer une.

L'employé va consulter ensuite le *fichier des stocks,* pour vérifier si la commande peut être honorée, si les stocks sont suffisants. Il doit soustraire sur cette fiche les

quantités à livrer, ou mettre en attente la commande, ou ne faire qu'une livraison.

Pour calculer la *facture,* il lui faut consulter la *table des prix*, y appliquer le taux de remise, effectuer un certain nombre d'opérations (multiplier le prix unitaire par la quantité livrable, additionner les résultats, y appliquer un taux de T.V.A.).



IL PEUT ALORS COMMENCER SON BON DE LIVRAISON

Notre employé doit alors *mettre à jour* la fiche client, la reclasser dans son bac, mettre à jour l'état des stocks, reclasser les fiches de chaque produit, regrouper les commandes non exécutées et ... *transmettre* le bon d'expédition et la facture au service responsable.



IL Y A LITIGE

Cet exemple nous montre comment un employé accomplit sa tâche quotidienne, à condition d'avoir au préalable, reçu ou élaboré des instructions de travail.

L'ensemble des instructions successives constitue une *chaîne d'instructions* qui représente le *programme de travail.* Ce *programme* serait consigné sur une fiche de la façon ci-contre.

PROGRAMME DE TRAVAIL

1 PRENDRE UNE COMMANDE DANS LE BAC DE GAUCHE

2 LIRE LE NOM DU CLIENT ET CONSULTER SA FICHE.

3 SI LA FICHE N'EXISTE PAS EN CRÉER UNE.

4 SI LA FICHE INDIQUE UN COMPTE NON A JOUR, NE PAS LIVRER ET LE SIGNALER AU SERVICE DES LITIGES.

5 SI LA FICHE INDIQUE UN COMPTE A JOUR, CONSULTER L'ETAT DES STOCKS DU PREMIER PRODUIT A LIVRER

6 SI LE STOCK EXISTE, COMMENCER LE BON DE LIVRAISON ET LA FACTURE. DÉDUIRE LA QUANTITÉ DU STOCK EXISTANT.

7 S'IL Y A RUPTURE DE STOCK, L'INDIQUER SUR LA COMMANDE, ET PRÉVOIR UNE LIVRAISON ULTÉRIEURE.

8 SI LE STOCK NE PERMET QU'UNE LIVRAISON PARTIELLE, L'INDIQUER SUR LE BON DE LIVRAISON ET SUR LA COMMANDE. PRÉVOIR UNE LIVRAISON ULTÉRIEURE.

9 POUR LE PRODUIT SUIVANT RETOURNER AU PARAGRAPHE 6.

10 S'IL N'Y A PAS D'AUTRE PRODUIT A LIVRER, COMMENCER LE CALCUL DE LA FACTURE.

11 CALCULER LES PRIX, REMISES, TAUX DE T.V.A.

12 COMPLÉTER LA FICHE CLIENT.

13 CLASSER LES COMMANDES EN DIFFÉRENTS LOTS :

_ ÉXÉCUTÉES

_ PARTIELLEMENT ÉXÉCUTÉES

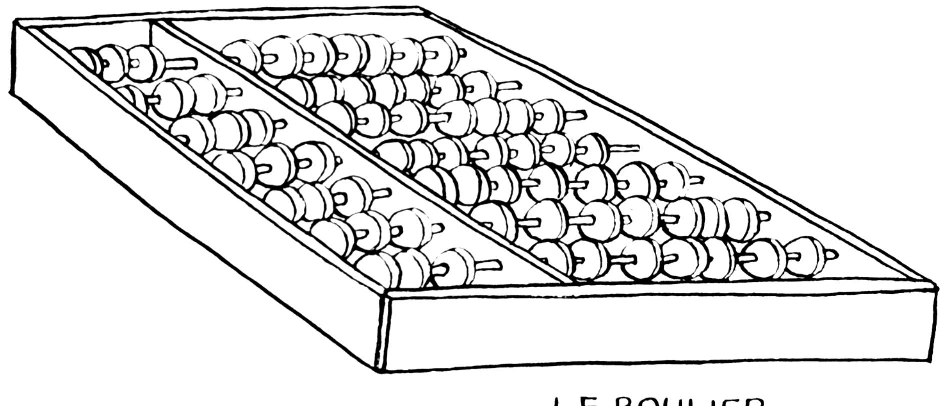
_ NON EXECUTÉES

14 TRANSMETTRE FACTURES ET BONS DE LIVRAISONS AUX SERVICES CONCERNÉS.

Un tel traitement se décompose en deux types d'opérations de natures différentes :

- les *opérations à caractère logique* que sont les recherches, comparaisons, vérifications, classement ...
- les *opérations de calcul* que sont les additions, multiplications ...

Pour éviter ces opérations dont la répétition est fastidieuse, l'homme a conçu des machines capables de les effectuer. Les premières d'entre-elles étaient tout justes capables de l'aider dans ses calculs. L'ancêtre de ces machines est le boulier.



LE BOULIER

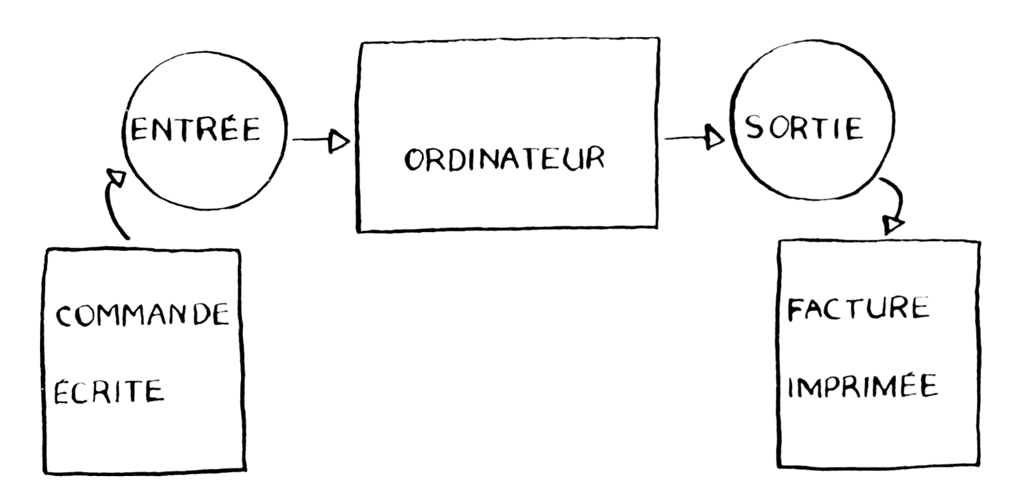
Puis les opérations logiques ont pu être appréhendées grâce à la technologie des ordinateurs. Peu nous importe de retracer ici l'historique de ces machines. Bornons-nous à comprendre le fonctionnement des systèmes actuels.

Notre intention est d'automatiser la facturation. Nous désirons donc entrer des commandes et recueillir en sortie une facture.

Cela suppose que l'on soit capable :

- *d'entrer la commande.* Il faut passer d'un document écrit à un travail sur machine. Ce travail est effectué par un *organe d'entrée*
- *d'en effectuer le traitement.* Une succession d'instructions de travail reprenant chaque opération à effectuer doit indiquer à la machine la marche à suivre. Les opérations arithmétiques et logiques sont effectuées par l'*organe de calcul et de commande.* L'ensemble des instructions nécessaires au fonctionnement sera élaboré sous la forme d'un *programme de travail*
- *de consulter le catalogue* des prix, le *fichier* des clients pendant le traitement. Ces données doivent avoir été stockées auparavant dans l'ordinateur : elles sont en *mémoire*
- *de restituer le résultat* du traitement sous une forme utilisable, c'est le travail de *l'organe de sortie.*

Le schéma global de ce traitement peut se représenter ainsi :



Ce chapitre nous a permis de définir les mots qui reviendront souvent au cours de ce livre :

INFORMATIQUE

FICHIER

PROGRAMME

TRAITEMENT

INSTRUCTIONS

CHAINE D'INSTRUCTIONS

ORGANE D'ENTREE

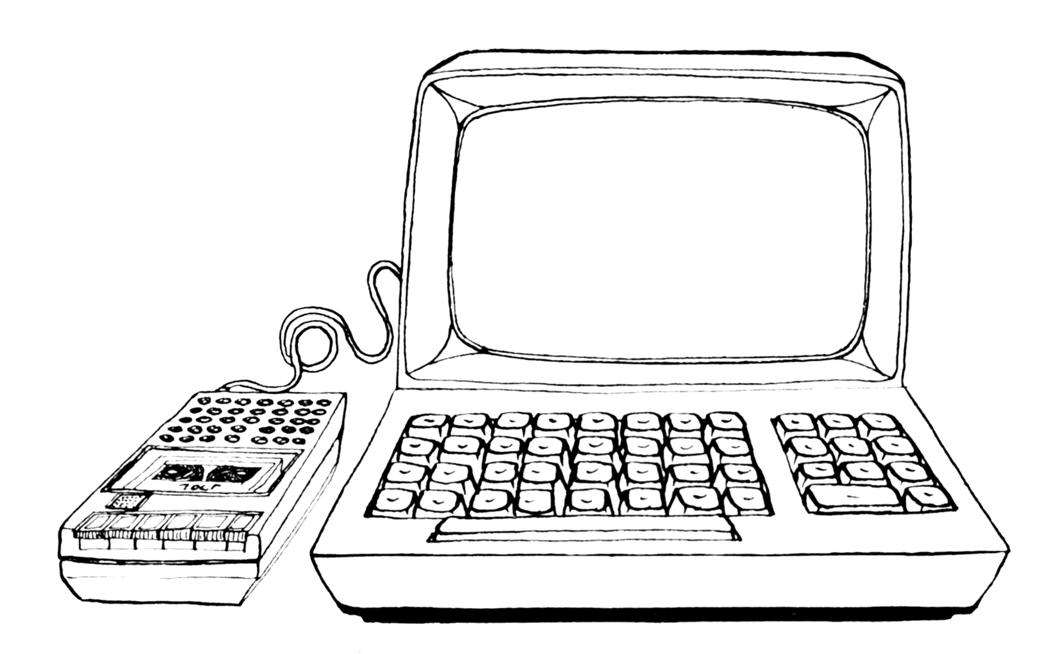
MEMOIRE

ORGANE DE CALCUL

ORGANE DE SORTIE



LES ORDINATEURS INDIVIDUELS



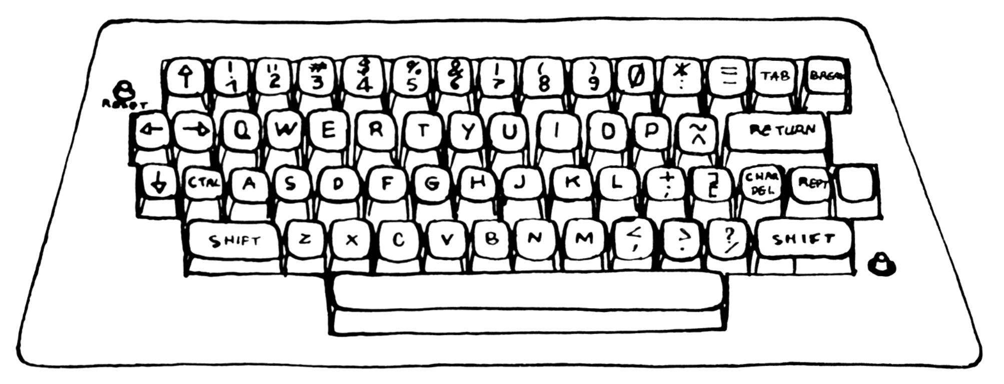
En analysant le travail d'exécution d'un employé et la prise en charge des opérations par un système informatique, nous avons défini les fonctions et organes essentiels d'un ordinateur. Nous allons, dans ce chapitre, faire connaissance avec ces différents organes.

L'Ordinateur individuel classique comporte 3 éléments :

Tout d'abord un *écran*, type écran de télévision, qui peut sur certains modèles être en couleur.

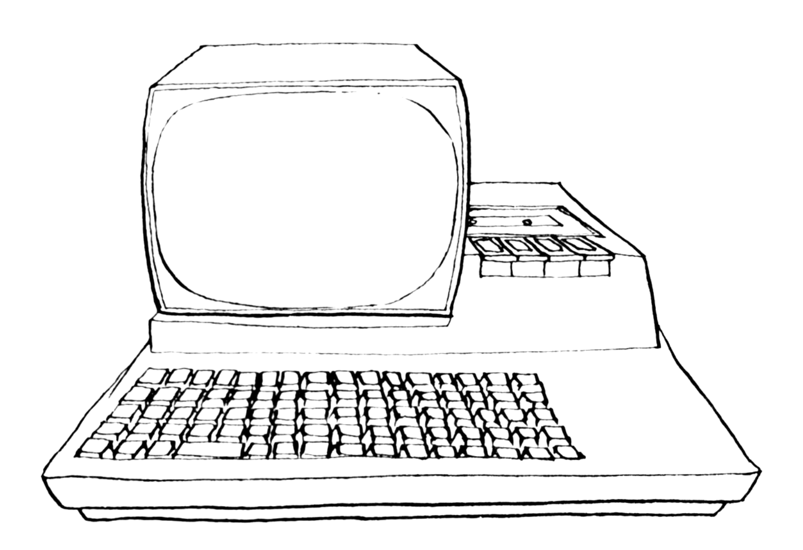
Un *clavier* analogue à celui d'une machine à écrire. Il comporte :

les lettres de l'alphabet ;
les dix chiffres de 0 à 9 ;
les symboles (* : + - ? !) ;
des signes pouvant servir au dessin ;
des touches spécialisées telles que : RUN-STOP-RETURN.



Intégré dans la console ou relié par un câble électrique, nous trouvons un *lecteur de cassette* absolument identique à ceux utilisés pour la reproduction musicale, avec les touches de fonctionnement :

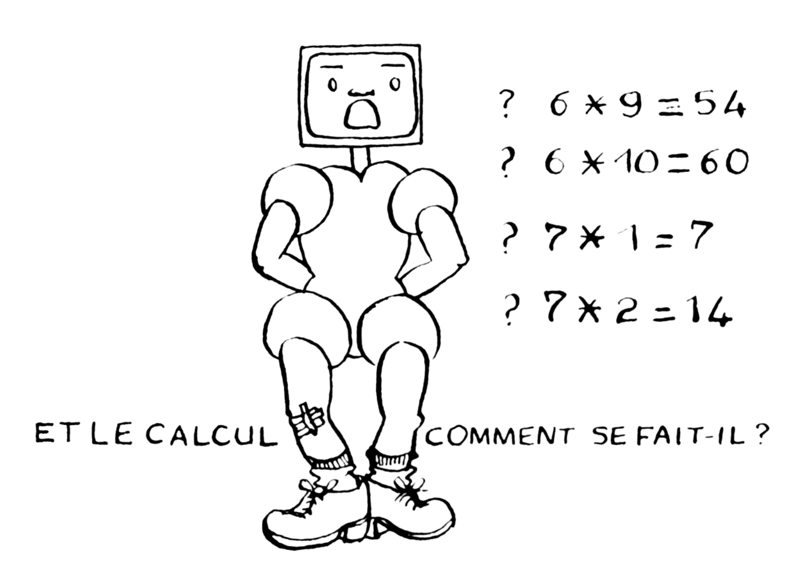
PLAY	*RECORD*	*REWIND*	*STOP*
MARCHE	ENREGISTRE	ENROULE	ARRET



Que se passe-t-il lorsque nous appuyons sur l'une des touches du clavier ? Nous faisons apparaître sur l'écran les caractères qui y sont représentés. Nous pouvons écrire un texte, tout comme sur une machine à écrire : certaines touches permettent le retour en arrière, l'espacement, le retour à la ligne, l'effacement, etc.

Nous pouvons ainsi remplir notre écran comme une feuille de papier, avec une différence : en bas de page, nous continuons à écrire mais ... la première ligne du texte, elle, a disparu ; notre écran ne nous permet donc pas de conserver le texte en entier, au contraire d'une feuille de papier:

Bien entendu, il est possible de mémoriser un texte, il suffit d'en donner l'instruction à l'ordinateur, faute de quoi il n'en fera rien. Nous verrons cet aspect dans le chapitre réservé à la programmation.



Et le calcul, comment se fait-il ? Si nous frappons 6 * 9, cette expression s'inscrit sur l'écran, mais nous n'obtenons aucun résultat. De fait, il est possible d'obtenir un résultat à condition de frapper :

? 6*9

Puis d'appuyer sur la touche RETURN ou ENTER sur d'autres modèles.

L'écran nous renvoie la réponse : **54**

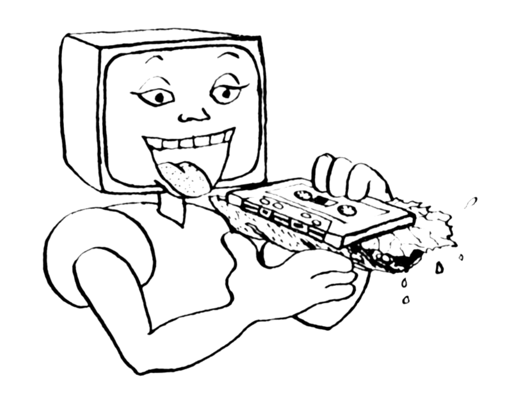
Remarquons, au passage, que *la multiplication s'exprime par* * et non par x, ceci afin de ne pas confondre avec x ou + .

Pour reprendre le schéma proposé au chapitre précédent : l'organe d'entrée est le clavier, l'organe de sortie est l'écran.

En fait, le traitement d'une commande ne pourra être réalisé que si l'ordinateur reçoit des instructions regroupées sous la forme d'un programme de travail.

Ce programme, ainsi que : le fichier client, la table des prix, le fichier des stocks, doivent être mémorisés sur la cassette, qui livrera son contenu à l'ordinateur pour le traitement nécessaire.

On peut ainsi disposer de *nombreux programmes enregistrés sur une ou plusieurs cassettes,* permettant à un ordinateur de résoudre différents problèmes.



LA CASSETTE PERMET LA MEMORISATION DES PROGRAMMES

La description ci-dessous, concerne un petit équipement d'Ordinateur Individuel avec ses avantages et ses inconvénients.

En *entrée,* nous pouvons agir par l'intermédiaire d'un *clavier*, ou d'une cassette contenant des informations déjà mémorisées.

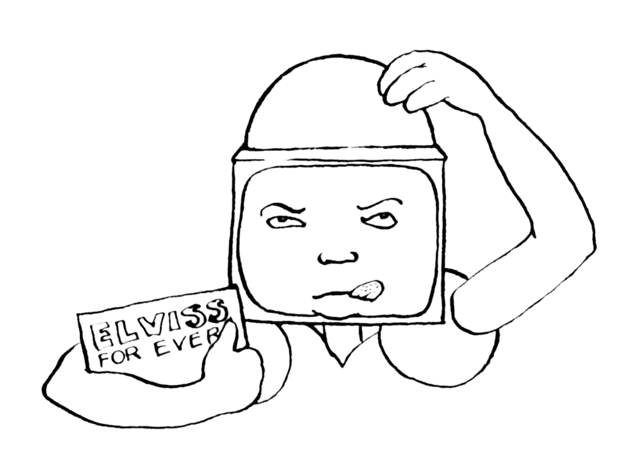
Le clavier permet d'agir au coup par coup, à une vitesse très lente, la nôtre, et c'est une source d'erreurs.

La *cassette* permet la *mémorisation des programmes,* mais pour atteindre une information placée à un bout, il faut la dérouler entièrement. L'*accès est dit «séquentiel».*

Le *disque* a été une amélioration substantielle en informatique car il a permis d'accéder plus facilement à une information. De même qu'il est plus aisé de sélectionner un morceau musical sur un disque que sur une bande magnétique, il est plus facile d'accéder à une information mémorisée sur un disque que sur une bande, le temps d'accès étant beaucoup plus bref. *L'accès est dit direct.*

En sortie, l'Ordinateur Individuel permet de *lire la réponse sur l'écran.* Ce mode de communication direct est d'un usage très commode, mais comme pour les informations d'entrée, les réponses peuvent être perdues. Aussi des équipements spécialisés peuvent les récupérer en sortie.

Le premier d'entre-eux est l'*imprimante*, qui frappe en sortie les textes et résultats de calcul, suivant un format préétabli. Ainsi une facture pourra être effectuée directement, sans intervention manuelle, avec la présentation voulue. Les capacités de ces machines à écrire sont déjà impressionnantes, puisque la vitesse de frappe est égale à plusieurs centaines de lignes à la minute.



LE DISQUE JOUE ALORS LE ROLE DE MEMOIRE EXTERNE A L'ORDINATEUR

Dans certains cas, il est inutile de conserver les informations sur un support de papier, c'est pourquoi on utilise aussi en sortie des *disques magnétiques.* La vitesse d'enregistrement est de l'ordre de quelques centaines de milliers de caractères par seconde et les possibilités atteignent des capacités de stockage de quelques millions de caractères par disque. Le disque joue alors le rôle de *mémoire externe à l'ordinateur.*

Lorsque l'on se réfère à ces capacités, il est bon de prendre un point de comparaison, le texte de ce livre va nous y aider :

une ligne de texte	60 caractères	
une page	55 lignes	= 3 300 caractères.

Un *disque* moyen de 2,5 millions de caractères contient quelque 800 pages de texte. Il s'agit là d'un stockage possible sous forme compacte.

La *disquette,* qui ne contient qu'un maximum de 1 000 000 de caractères, est d'un usage fréquent pour l'utilisation individuelle.

Bien entendu, il est possible de faire jouer au disque le rôle d'organe d'entrée à la place du clavier, dans la mesure où l'ordinateur lui fait appel ; par exemple dans notre modèle de facturation, les divers fichiers (client, prix, stocks) pouvant être mémorisés sur un ou plusieurs disques.

L'ordinateur proprement dit est un assemblage électronique, le cœur et le cerveau de cet assemblage sont appelés microprocesseur. A cet assemblage nous demandons de réaliser : des calculs, des comparaisons, le tri et le classement des informations.

Ceci est réalisé si nous lui en donnons l'*instruction.* Un ensemble d'instructions constitue un *programme.*

Plusieurs programmes permettent de faire fonctionner l'ordinateur. On leur donne l'appellation de *logiciel* ou de *software* en anglais.

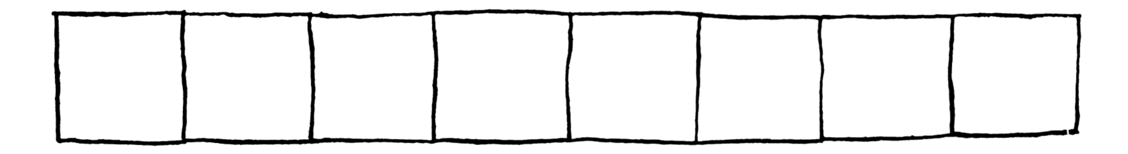
L'ordinateur devra, à partir des éléments fournis en entrée, effectuer le travail et restituer en sortie des informations utilisables.

Sans entrer dans des détails complexes, nous pouvons considérer que l'ordinateur est composé d'un grand nombre de petites cases.

Ces cases sont regroupées par blocs de 8 composants, ce que l'on appelle un *octet* (byte en anglais). Un octet peut contenir un caractère, un chiffre, un signe. La quantité d'octets détermine la capacité de l'ordinateur. Elle s'exprime en milliers d'octets ou *Kilo-octets* (par analogie à Kilogramme). Un souci d'abréviation a amené les constructeurs à parler de *K*.

1 K = 1024 Octets = 8192 éléments binaires appelés BIT
(contraction de BInary Digit)

Les derniers développements de l'électronique ont permis de réaliser ces éléments sous une forme très compacte grâce à l'utilisation des circuits intégrés.



REPRESENTATION D'UN OCTET

Les petits ordinateurs individuels ont des capacités de 8K, 16K ... Les très gros ordinateurs atteignent 512K, 1024K, mais ils ne peuvent être considérés comme individuels.

Munie de ces atouts, la machine doit pouvoir :

- □ Mémoriser des informations, grâce à un élément interne de la machine appelé *mémoire interne ;*
- □ Calculer, traiter les informations, grâce à un élément appelé *organe de calcul et de traitement.*
- □ Contrôler son propre fonctionnement, grâce à un *organe de contrôle.*

MEMOIRE INTERNE	
CALCUL ET TRAITEMENT	ORGANE DE CONTROLE

ORGANISATION DE LA PARTIE CENTRALE D'UN ORDINATEUR

Chacune de ces parties de l'ordinateur est composée de milliers d'octets. Au cours du fonctionnement, un passage continuel d'informations se produit entre la mémoire et l'organe de calcul, puis de cette information à la mémoire.

Prenons un exemple simple. Il me faut multiplier 3 par 2 ou 3 * 2.

Le stockage de 3, de * et de 2, a été fait en mémoire, dans des octets. Les contenus de ces derniers passent dans l'organe de calcul qui trouve la valeur 6 et remet en mémoire cette valeur. Nous aurons l'occasion au chapitre suivant de revenir sur cette notion.

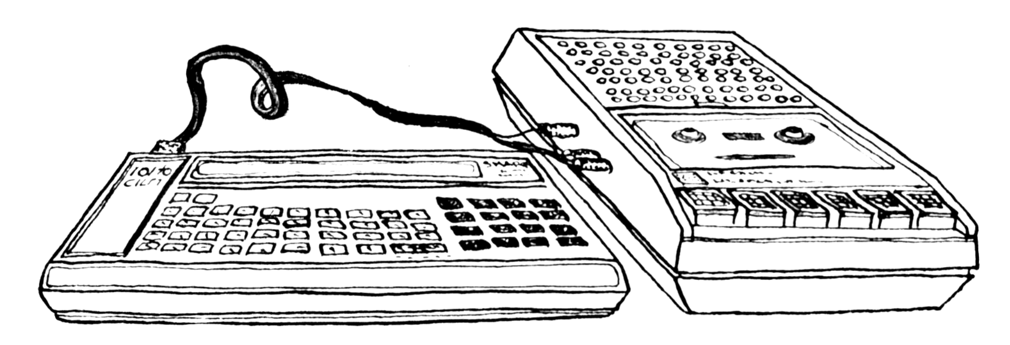


Les équipements tels que les écrans, les imprimantes, qui sont extérieurs à l'ordinateur, sont appelés équipements *périphériques.*

Les efforts des constructeurs se sont portés vers la réalisation de systèmes de petites dimensions. C'est ainsi que plusieurs sociétés ont réalisé des modèles «de poche», fonctionnant comme un ordinateur classique. L'écran de ces ordinateurs fait apparaître une seule ligne de caractères (24 par exemple), et les dimensions du système hors-tout sont très faibles (par exemple : épaisseur 1,2 cm, largeur 7 cm, longueur 17,5 cm).

La conservation des programmes se fait soit en mémoire interne soit sur cassette, le lecteur de cassette étant l'appareil standard du commerce.

Bien entendu toutes les machines n'ont pas les mêmes possibilités, cependant la majorité des exemples choisis dans ce livre fonctionnent sur tous les ordinateurs individuels qu'ils soient de table ou de poche.



UN ORDINATEUR DE POCHE

Pour utiliser ces divers matériels, il faut offrir à notre ordinateur des programmes de travail (ou logiciel). Certains programmes existent dans le commerce : déjà réalisés, ils sont vendus enregistrés sur des cassettes et il est possible à tout acheteur de les utiliser directement (mais attention, ils doivent avoir été conçus pour le matériel dont on dispose), on les appelle alors *progiciels* (contraction de *pro*duits lo*giciels*).

Ce chapitre nous a amené à mettre en jeu un certain nombre de *matériels.* Les Américains parlent de *hardware* qui signifie «quincaillerie», c'est en fait, tout ce qui a trait au matériel. Nous avons évoqué successivement :

CLAVIER	OCTET (BYTE)-KILO OCTET-K
ECRAN	MEMOIRES
CASSETTE	ORGANE DE CALCUL ET DE TRAITEMENT
DISQUETTE	ORGANE DE CONTROLE
DISQUE	PROGRAMMES
IMPRIMANTE	

Nous allons voir maintenant comment réaliser soi-même de petits programmes. Si vous disposez d'un ordinateur individuel, vous tirerez un très grand profit de son utilisation simultanée à la lecture. Faites-vous simplement expliquer les quelques commandes nécessaires à l'utilisation du langage BASIC.

PROGRAMMES ET LANGAGES



Au cours des chapitres précédents, nous avons été amenés à décrire ce qu'est un programme. Le premier que nous ayons rencontré, celui de l'employé, se présentait comme une suite d'instructions, que nous avons numérotées dans l'ordre de leur réalisation (voir le chapitre I).

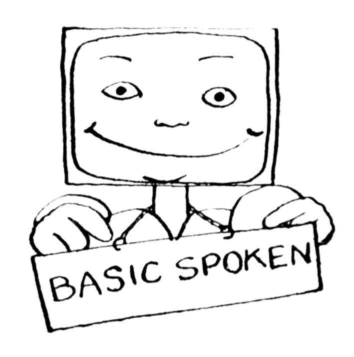
Le programme de travail d'un ordinateur est une succession d'instructions écrites. Chaque instruction porte un numéro qui indique l'ordre dans lequel elle doit être exécutée.

Lorsque nous écrivons ces instructions, nous ne sommes pas sûrs de ne pas en oublier et il est commode de les numéroter de 5 en 5, de 1Ø en 1Ø, ou à toute autre cadence, ce qui permet d'intercaler facilement une nouvelle ligne en l'affectant d'un numéro intermédiaire.

L'on écrit ainsi un programme en numérotant :

5 Première instruction
1Ø Seconde instruction (afin de ne pas confondre «O» et zéro, ce dernier est écrit Ø)
15 Troisième instruction
2Ø Quatrième instruction

Chaque instruction est entrée en machine lorsque l'on a appuyé sur la touche [RETURN] ou [↵] ou [ENTER] , suivant les matériels.



Il est tout à fait possible d'y ajouter une instruction de la façon suivante :

8 Nouvelle instruction

Oui, mais, direz-vous, elles ne sont plus dans l'ordre ! Qu'à cela ne tienne, frappons Liste sur le clavier puis Return (ou Enter...), l'ordinateur affiche les instructions dans l'ordre des numéros croissants, il a de lui-même placé les instructions au bon endroit. N'est-ce pas sympathique ?

Nous venons d'utiliser une première instruction qui a fait exécuter un travail par l'ordinateur : l'affichage des lignes du programme contenu dans sa mémoire.

Ces instructions doivent être libellées suivant un code préalablement établi, les informaticiens disent qu'elles constituent des *langages,* et les ont baptisés de noms tels que : *Fortran, Cobol, Algol, et ... Basic.*

Chaque langage a des inconvénients et des avantages qui en déterminent l'usage. La plupart des grands constructeurs étant américains, ces instructions sont formulées en anglais. Mais en 1971, la revue *L'Ordinateur Individuel* a eu l'idée d'utiliser un langage écrit en français, qui est une traduction du *Basic* et, avec humour l'a appelé le *Basicois.*

C'est ce dernier qui va nous servir au cours des chapitres suivants pour la rédaction des programmes. Bien entendu, si vous possédez un ordinateur, il vous faudra le faire fonctionner en Basic, à moins que vous n'utilisiez la cassette Basicois éditée par les auteurs de *L'Ordinateur Individuel.*

Nous rédigerons donc les programmes en Basicois et indiquerons, en italique, les instructions Basic correspondantes, selon le principe du sous-titrage.

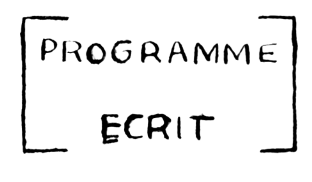
LISTE EN BASICOIS
LIST EN BASIC

Le Basicois, ou le Basic, nous permettra donc, à partir d'instructions frappées sur un clavier, enregistrées sur cassette, ou sur mini-disquette, de faire fonctionner l'ordinateur. Les instructions sont *des ordres,* elles sont exprimées au *mode impératif !.*

Mais attention, un langage est quelque chose de précis, tout point, toute virgule a son importance !...

L'instruction EXE *(RUN)*

Pour faire exécuter un programme, nous devons utiliser l'instruction **EXE** (exécute). Au cours des chapitres à venir, nous la présenterons entre le programme et l'apparition de ce que nous verrions sur l'écran. Ceci indique qu'il faut frapper **EXE** pour obtenir l'exécution du programme.

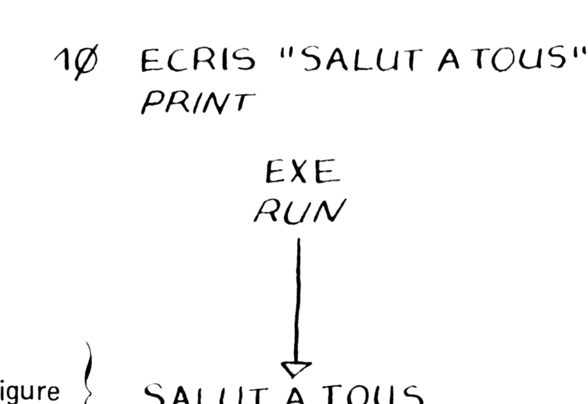


(nous écrivons au clavier)



L'instruction ECRIS *(PRINT)*

Si nous écrivons : PROGRAMME III-1



L'ordinateur écrit sur l'écran ce qui figure entre guillemets

Attention l'instruction ECRIS concerne les mots compris entre les guillemets et seulement ceux-là. *Les guillemets sont obligatoires.*

Pour un tableau, nous pouvons nous organiser de façon à avoir une mise en page agréable, pour cela nous pouvons programmer :

PROGRAMME III-2

```
10  ECRIS "NOM", "PRENOM"
    PRINT
```

EXE
RUN

```
NOM          PRENOM
```

La virgule a permis d'espacer deux écritures, aérant ici suffisamment le texte. Cette instruction permet de réaliser la composition d'un texte un peu comme on le ferait avec la tabulation sur une machine à écrire. Elle permet aussi d'autres possibilités que nous verrons plus tard.

EXERCICE III-1

Quelles instructions faut-il écrire pour obtenir à l'exécution :

```
NOM

PRENOM
```

La solution des exercices est en fin de livre Annexe I, page 79.

EXERCICE III-2

Que donne en exécution ?

```
10 ECRIS "NOM, PRENOM"
```

L'instruction DEMANDE *(INPUT)*

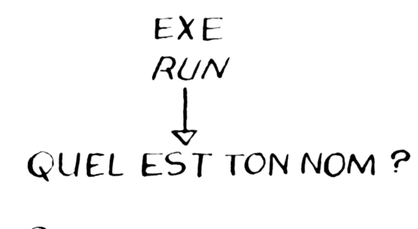
Nous voulons que l'ordinateur pose une question, par exemple quel est le nom de l'opérateur, et qu'il enregistre ce nom.

Comme nous ne connaissons pas la réponse, avant que l'opérateur ait répondu, nous désignons dans le programme ce nom par **A$**, de la façon suivante :

PROGRAMME III - 3

10 ECRIS "QUEL EST TON NOM ?"
PRINT

20 DEMANDE A $
INPUT



?

La réponse est à taper au clavier et est suivie d'une frappe sur la touche RETURN ou ⮐ ou ENTER, suivant les matériels.

L'ordinateur pose une question et signale par ? qu'il attend une réponse qui lui a été désignée par **A$**. La réponse dépend de l'opérateur et **A$** est une *variable.*

Le symbole **$** signifie que l'on attend une réponse qui soit une suite de caractères quelconques. Si l'on avait écrit A comme variable, la réponse devait impérativement être un nombre.

Supposons que la réponse que vous écrivez par l'intermédiaire du clavier soit **DURAND**, cette réponse est inscrite en mémoire et l'ordinateur la conservera et la restituera chaque fois que l'on réclamera A$.

Par exemple en ajoutant l'instruction 30

30 ECRIS A $
PRINT



Cette instruction ECRIS sans guillemets n'applique pas la règle définie plus haut ! Si nous avions respecté cette règle en écrivant :

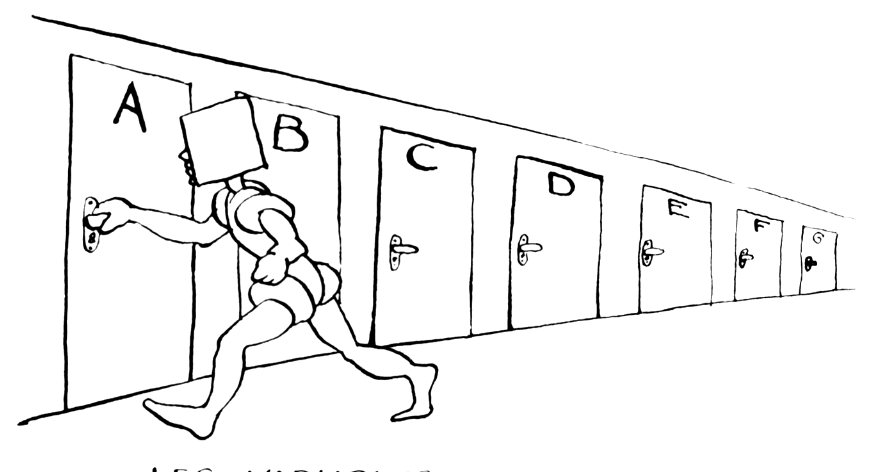
30 ECRIS "A $"
PRINT

Nous aurions obtenu à l'exécution :

A$

Or, ce n'est pas le but recherché : nous voulons appeler quelque chose enregistré dans la mémoire et que nous avons appelé A$. L'*absence de guillemets* indique à l'ordinateur que ce qui suit l'ordre ECRIS n'est pas ce qu'il faut imprimer, mais le nom de l'endroit où se trouve ce qu'il faut imprimer.

Ces «endroits» sont les *variables.*



LES VARIABLES

La notion de variable qui vient d'être introduite est très importante car sur elle repose toute la programmation.

Pour bien comprendre son effet, reprenons quelques éléments du chapitre II.

Nous avons précédemment imaginé que l'ordinateur était composé de petites cases, huit d'entre-elles constituent un octet. Le regroupement de plusieurs de ces octets constitue un bloc auquel on attribue l'étiquette A (ou B ou C ...) avec ou sans le symbole $ suivant que l'on attend une information constituée de chiffres ou de caractères.

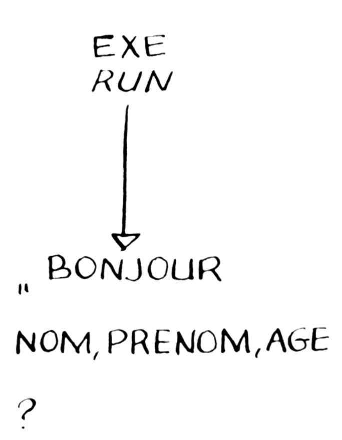
Cet étiquetage est réalisé par le programme et lorsque nous appelons la variable A, l'ordinateur exécutera son traitement en utilisant la valeur contenue dans le bloc désigné par A.

Il est possible de se constituer par programme une quantité importante de variables, puisqu'il est possible d'en constituer par utilisation d'une lettre ou par combinaison de deux lettres, avec ou sans le symbole $. Appliquons immédiatement cette possibilité.

Le programme que nous envisageons est plus inquisiteur que le précédent puisqu'il va questionner l'opérateur sur son nom, présom, âge, et se présente ainsi :

PROGRAMME III-4

```
10 ECRIS "BONJOUR"
   PRINT
20 ECRIS "NOM, PRENOM, AGE"
   PRINT
25 DEMANDE A$, B$, C
   INPUT
```



Vous devez ici répondre. *L'ordinateur attribuera chacune de vos réponses aux trois variables* que vous avez indiquées et *dans l'ordre choisi.*

Sur l'écran, apparaissent successivement les réponses :

```
BONJOUR
NOM, PRENOM , AGE
? DURAND
? JACQUES
? 39
```

Nous remarquons au passage que la troisième réponse est 39 et non «Trente Neuf» ans, puisque nous avions introduit une *variable C* non suivie au symbole **$**

qui attend une réponse composée de *chiffres uniquement.*

Nous disposons maintenant d'informations sur l'opérateur et nous pouvons les utiliser comme nous le souhaitons. Essayons une présentation sur deux lignes grâce à deux nouvelles lignes de programme :

```
30  ECRIS A$,B$
    PRINT

35  ECRIS C;"ANS"
    PRINT

        EXE
        RUN
         |
         v
    DURAND  JACQUES

    39 ANS
```

Cet exemple simple utilise trois variables que nous avons su utiliser, bien entendu, il ne s'agit là que d'un début, nous ferons mieux par la suite.

Notez également que nous progressons dans l'utilisation de l'instruction ECRIS. En ligne 30 nous citons deux variables, nous *pouvons donc citer après un ordre ECRIS plusieurs variables* dont les contenus seront affichés successivement. En mettant une virgule les contenus affichés seront tabulés, en mettant un point-virgule ils seront mis bout à bout.

C'est le cas de la ligne 35 où, nouvelle innovation, sont mélangés un nom de variable et un libellé entre guillemets. *Dans une instruction ECRIS libellés et noms de variables peuvent être mélangés.*

Autres utilisations de DEMANDE ***(INPUT)***

Au lieu d'écrire :

```
5   ECRIS "QUELEST TON NOM ?"
    PRINT

10  DEMANDE A$
    INPUT
```

Il est possible de remplacer sur la grande majorité des matériels ces deux instructions par une seule :

```
5    DEMANDE "QUEL EST TON NOM ?"; A$
     INPUT
```

Cette instruction a le même effet que les deux précédents, mais attention à la position de [;] .

Au cours des chapitres suivants nous séparerons toujours ECRIS *(PRINT)* et DEMANDE *(INPUT)*, de façon à bien montrer qu'il y a deux actions différentes.

EXERCICE III-3

Compléter le programme suivant :

```
10   ECRIS "MARQUE DU VEHICULE"
     PRINT

15

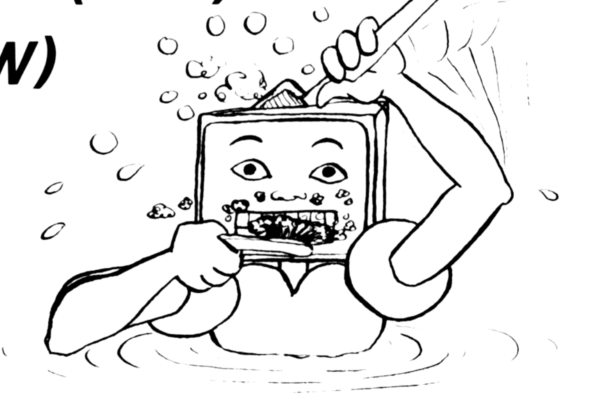
20   ECRIS "N° D'IMMATRICULATION"
     PRINT

25

30   ECRIS
     PRINT
```

Les instructions FIN *(END)* et NETTOIE *(NEW)*

Il est indispensable de prévenir la machine du début et de la fin du programme. L'instruction **FIN** arrête le déroulement du programme, l'instruction **NETTOIE** permet de nettoyer la mémoire de la machine afin de ne pas introduire d'instructions parasites dans le déroulement.



L'INSTRUCTION NETTOIE *(NEW)*

Cette dernière est utilisée chaque fois que l'on veut employer un nouveau programme. Cette instruction a le même effet que le chiffon pour nettoyer la craie sur un tableau noir.

Le programme précédent s'achèvera donc par la ligne :

```
4Ø FIN
   END
```

L'instruction FAIS *(LET)*

Cette instruction sert à *affecter* à une variable située à gauche du signe = ce qui est indiqué à droite de ce signe.

Ce peut-être un nombre, le contenu d'une autre variable ou le résultat d'un calcul portant sur des nombres ou sur le contenu de plusieurs variables. Ainsi : (*)

```
5  FAIS A = 1Ø
   LET
```

Affecte à A le contenu 10. Si nous donnons l'instruction :

```
6  ECRIS A
   PRINT
```

Nous obtenons la réponse :

```
1Ø
```

L'instruction suivante :

```
1Ø FAIS B = A * 5
   LET
```

affecte à B le contenu A * 5.

Si nous donnons l'instruction :

```
11 ECRIS B
   PRINT
```

nous obtenons la réponse : 5Ø (puisque 5Ø = 1Ø * 5)

(*) N'oubliez pas de frapper NETTOIE (NEW) avant tout nouveau programme. Si vous ne le faites pas le résultat sera toujours surprenant !

Ajoutons les instructions :

15 FAIS C=B*A
LET

16 ECRIS C
PRINT

la réponse est : 5ØØ

car C= A*B=A*A*5=1Ø*1Ø*5

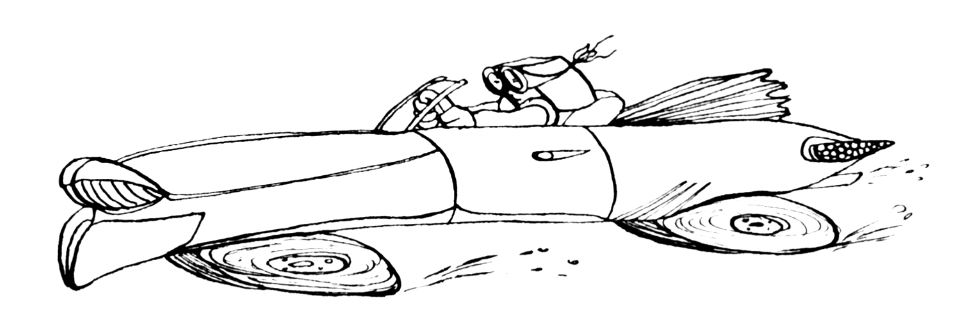
L'instruction :

15 FAIS C=B*A
LET

a pour effet d'attribuer à la variable C une valeur égale au produit de B par A.

Remémorons-nous l'image qui a été prise précédemment. L'ordinateur va :

- chercher la valeur de B dans un bloc étiqueté B ;
- puis la valeur de A dans un bloc étiqueté A ;
- multiplier ces deux valeurs
- placer la valeur de C dans un nouveau bloc.



Un exemple va nous aider à comprendre ce qui se passe. Nous avons une voiture qui consomme 9 litres aux 100 kilomètres. Quelle sera la consommation pour 462 kilomètres ?

Ce calcul est très simple, faisons-le exécuter par l'ordinateur. Il s'agit d'entrer la consommation aux 100 kilomètres, B, de calculer la consommation au kilomètre $\frac{B}{100}$ puis de multiplier ce résultat par la distance parcourue C.

Le résultat obtenu $A = \frac{B}{100} * C$ doit être affiché à l'écran.

Nous disposons de tous les éléments nécessaires à la réalisation du programme.

PROGRAMME III-6

```
5   ECRIS "QUELLE EST LA CONSOMMATION DE VOTRE
    PRINT
    VOITURE AUX 1Ø0 KM?"

1Ø  DEMANDE B
    INPUT

15  ECRIS "QUELLE DISTANCE AVEZ-VOUS PARCOURUE?"
    PRINT

2Ø  DEMANDE C
    INPUT

25  FAIS A = (B/1ØØ) *C
    LET

3Ø  ECRIS "VOUS AVEZ CONSOMME "; A ; "LITRES"
    PRINT

35  FIN
    END
```

EXE
RUN



```
QUELLE EST LA CONSOMMATION DE VOTRE VOITURE AUX 1ØØ KM ?

?9
```

QUELLE DISTANCE AVEZ-VOUS PARCOURUE ?

?462

VOUS AVEZ CONSOMME 41,58 LITRES

Attention à la présentation des calculs, la division est indiquée par la barre de fraction /. Pour s'y retrouver l'ordinateur respecte des règles. Si nous avions écrit A = B/100*C il aurait compris $A = \frac{B}{100*C}$ il faut donc soit mettre des parenthèses comme nous l'avons fait, soit écrire A = C*B/100.

Si vous désirez faire un autre calcul, il vous suffit de faire à nouveau EXE, et de rentrer les informations réclamées à nouveau.

EXERCICE III-4

Nous souhaitons obtenir le tableau suivant à partir du programme III-6. Que faire ?

CONSOMMATION AUX 100 KM	DISTANCE PARCOURUE	CONSOMMATION TOTALE
9	462	41.58

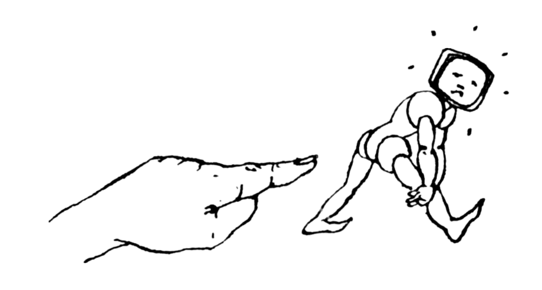
Mais, direz-vous, j'ai toujours la même voiture, donc elle consomme toujours 9 litres aux 100 Km, comment faire pour éviter de redonner cette information lorsque je veux effectuer plusieurs calculs sur des distances différentes ?

Une nouvelle instruction va nous y aider.

L'instruction VATEN *(GOTO)*

Il s'agit là d'une instruction à caractère original par rapport aux précédentes.

En effet nous avons dit que les instructions se déroulaient dans un ordre logique de numérotation croissante, et bien VATEN (GOTO) placée au cours du programme interrompt ce déroulement pour le replacer sur une autre instruction



L'INSTRUCTION VATEN (GOTO)

du programme, soit avant, si le numéro d'ordre est plus petit, soit après (numéro d'ordre plus grand).

Ainsi :

1ØØ VATEN 5Ø
GOTO

replace le déroulement du programme en 5Ø et fait exécuter les instructions de 5Ø à 1ØØ où il est renvoyé en 5Ø et ainsi de suite ...

Pour notre calcul de consommation, il suffit de compléter le programme précédent par :

35 VATEN 15
GOTO

Cette instruction permet après avoir effectué le calcul de repartir en 15 pour une nouvelle distance, le programme se déroule ainsi :



QUELLE EST LA CONSOMMATION DE VOTRE VOITURE AUX 1ØØ KM ?
?9

QUELLE DISTANCE AVEZ-VOUS PARCOURUE ?
?462

VOUS AVEZ CONSOMME 41,58 LITRES

QUELLE DISTANCE AVEZ-VOUS PARCOURUE ?
125

VOUS AVEZ CONSOMME 11,25 LITRES

L'ordinateur continue à vous poser des questions sur les distances parcourues autant

de fois que vous le souhaitez. Lorsque vous en aurez assez, il suffira de frapper sur la touche STOP. (Sur certains modèles ce sera la touche BREAK).



IL SUFFIRA DE FRAPPER SUR LA TOUCHE STOP.

EXERCICE III-5

Reprenant le programme III-6, que se passe-t-il si nous y introduisons :

soit : 27 VATEN 15
GOTO

soit : 27 VATEN 35
GOTO

A propos de FAIS *(LET)*

La plupart des systèmes informatiques suppriment :

l'injonction FAIS
LET

Ainsi : C = A * B

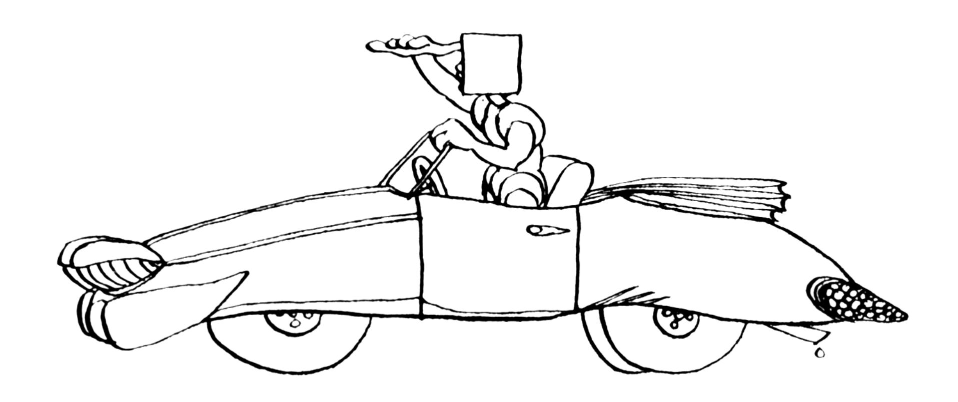
est identique à : FAIS C = A * B
LET

Cependant, nous conserverons cette bonne habitude tout au long des chapitres à venir.

Nous avons ici fait un bon bout de chemin. Ce chapitre nous a permis de voir un certain nombre d'instructions écrites en Basicois et *Basic.*

BASICOIS	*BASIC*
LISTE	*LIST*
EXE	*RUN*
ECRIS	*PRINT*
DEMANDE	*INPUT*
FIN	*END*
NETTOIE	*NEW*
VATEN	*GOTO*
FAIS	*LET*
ARRET	*STOP*

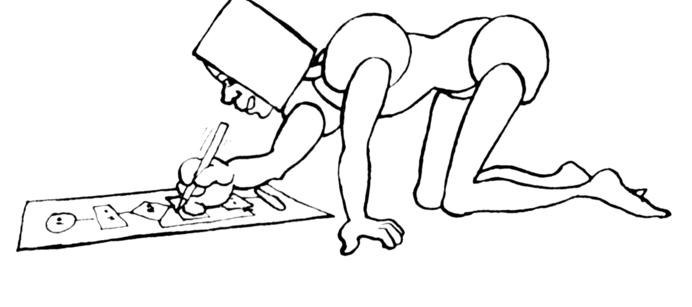
Nous avons aussi abordé la notion de *variables,* et réalisé de petits morceaux de programmes. Pour aller plus avant, nous allons faire appel à une technique d'*analyse* qui constitue un moyen commode de décortiquer un problème lorsque celui-ci devient plus complexe. Ce sera l'objet du chapitre suivant.



ANALYSER POUR PROGRAMMER

Nous avons déjà vu que l'écriture d'un programme nécessite une grande rigueur dans la formulation, tout point, toute virgule ayant en fait une signification qui leur est propre.

Avant d'écrire d'autres cas de programmation, il nous faut introduire une autre notion qui est la rigueur du raisonnement.



Lorsque le problème à résoudre devient quelque peu complexe, il est nécessaire d'en faire une *analyse* minutieuse et de n'aborder la programmation que dans une seconde phase.

Les problèmes que nous allons évoquer n'ont pas de grande complexité, et vont nous permettre d'aborder la phase d'analyse de façon très aisée.

Tout d'abord, effectuons un retour en arrière sur le problème de la consommation de votre véhicule (Programme III-5). Pour le résoudre, nous avons développé une démarche en cinq points, qui sont les suivants :

1 - Début du programme

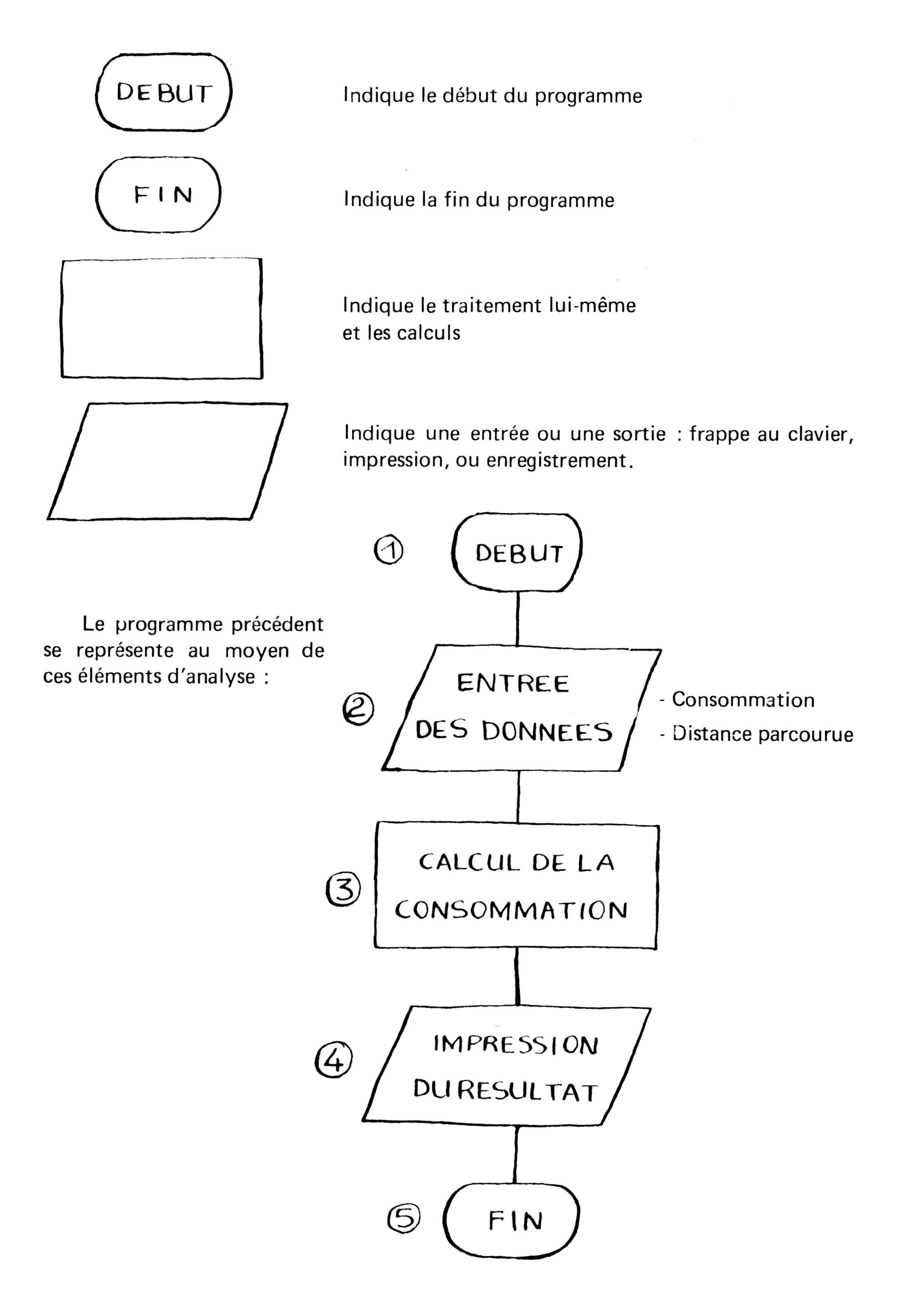
2 - Entrée des données : consommation aux 100 Km
distance parcourue

3 - Calcul de la consommation

4 - Impression du résultat

5 - Fin de programme

L'analyse ainsi faite se représente au moyen de quelques figures dont nous allons indiquer la signification.



Indique le début du programme

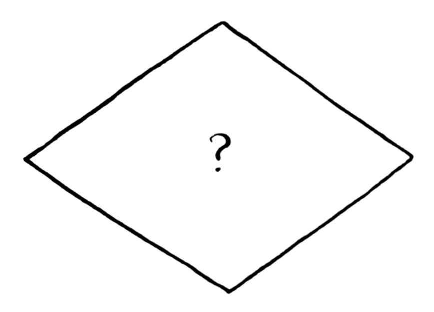
Indique la fin du programme

Indique le traitement lui-même et les calculs

Indique une entrée ou une sortie : frappe au clavier, impression, ou enregistrement.

Le programme précédent se représente au moyen de ces éléments d'analyse :

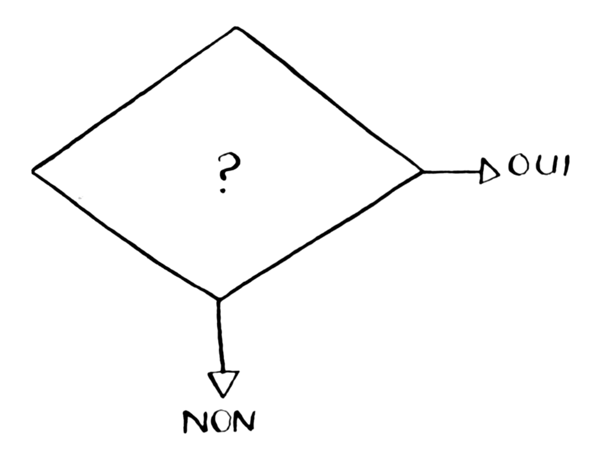
Pour compléter la panoplie des schémas utilisés, ajoutons-y :



Ce schéma signifie que l'on pose une condition à laquelle on peut avoir deux réponses, soit OUI, soit NON. Graphiquement :

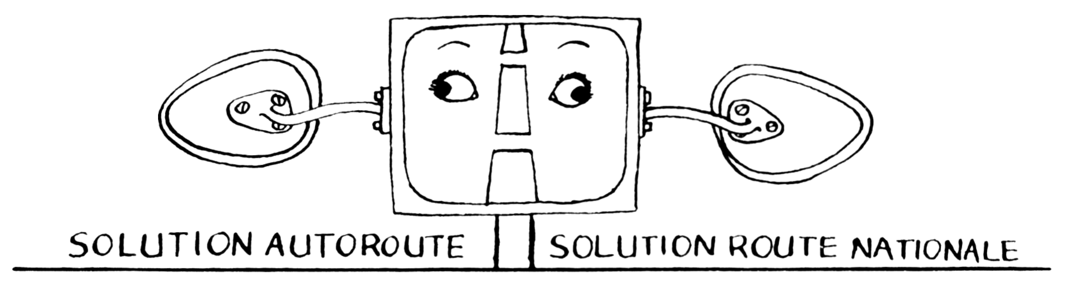
La réponse OUI implique une direction

La réponse NON implique une seconde direction



EXEMPLE IV-1

Nous devons aller de Paris à Bruxelles en voiture. Deux solutions sont possibles : l'autoroute et la route nationale. Quels seront le coût et la durée du trajet de chaque itinéraire ?



L'analyse de ce problème peut se formuler de la façon suivante :

1. Entrer les données : Prix de l'essence
Distance par Autoroute
Distance par Route Nationale
Vitesse et consommation dans chaque cas
2. Poser le choix initial : Route ou Autoroute ?

Solution Autoroute	Solution Route Nationale
3. Calculer le coût en essence	6. Calculer le coût en essence
4. Ajouter le coût du péage	
5. Calculer le temps nécessaire	5. Calculer le temps nécessaire

8. Imprimer la durée du trajet et le coût du voyage.

A cette analyse correspond un *organigramme*. Chaque case a reçu un numéro correspondant à l'analyse effectuée. *(voir organigramme ci-contre).*

Remarquons que notre schéma présente une nouveauté : nous avons utilisé un *branchement*. Pour le lire, il faut suivre les chemins matérialisés par les traits :

Le chemin OUI passe par (6) et (7)

Le chemin NON (solution autoroute) par (3) (4) et (5)

(8) et (9) sont communs aux deux situations.

Pour programmer, il nous manque une instruction correspondant à la condition posée, cette instruction est **SI** .... **ALORS.**

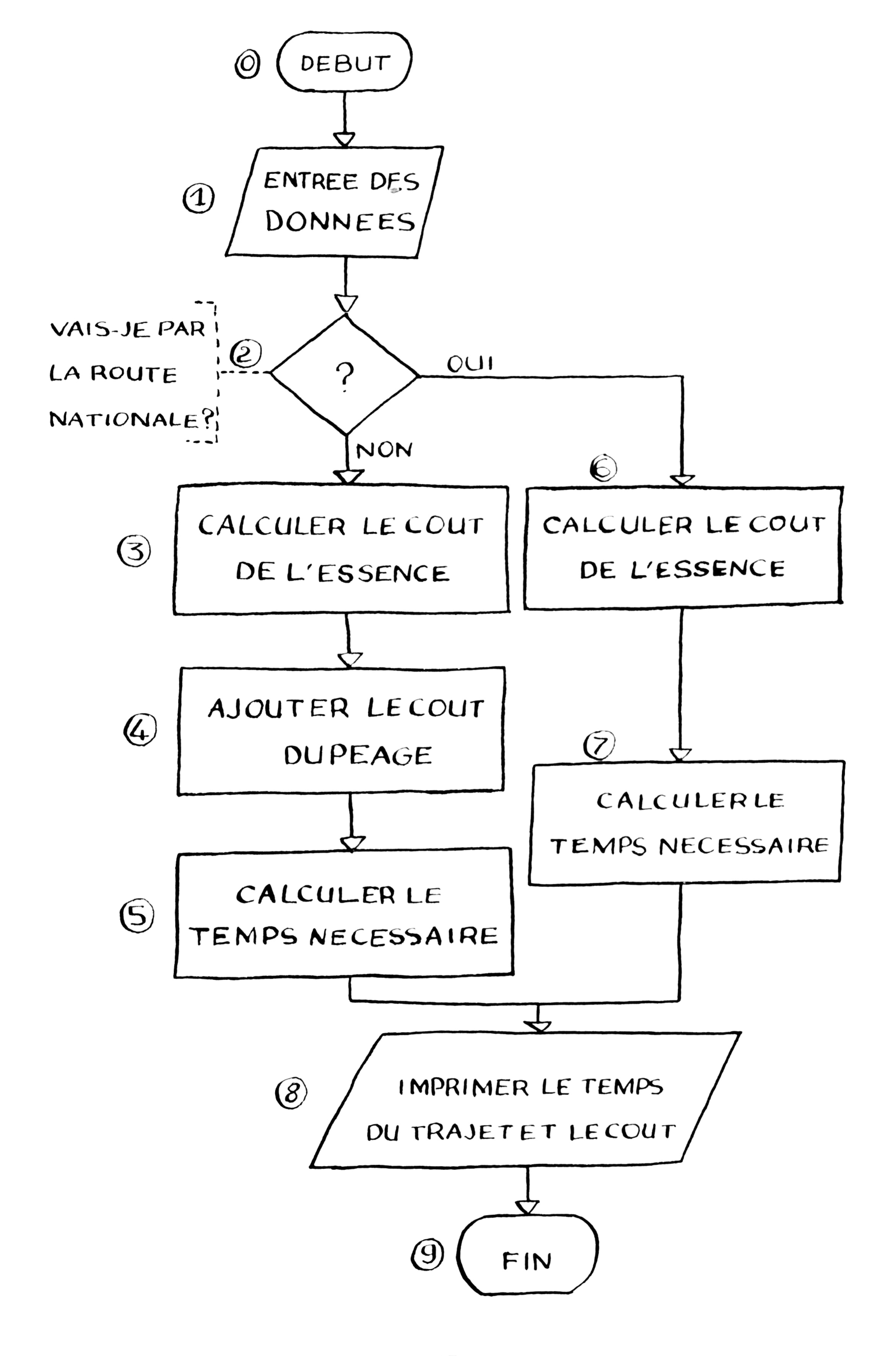
Si la réponse à la question «Vais-je par l'autoroute ?» est *OUI*, **alors** je calcule le coût de l'essence, du péage et le coût total.

Si la réponse est *NON*, je vais donc par la route nationale. La consommation est différente, le coût le sera aussi.

L'instruction : SI... ALORS... *(IF... THEN...)*

Cette instruction dite de *test de condition* permet d'énoncer une condition à laquelle deux réponses sont possibles OUI ou NON. Elle est écrite par comparaison de deux variables.

Ces variables peuvent représenter des nombres, mais aussi des caractères, tels que des lettres ou un texte. Dans ce cas, souvenons-nous que les variables sont suivies du signe $.


0
DEBUT
1
ENTREE DES DONNEES
VAIS-JE PAR LA ROUTE NATIONALE?
2
?
OUI
NON
3
CALCULER LE COUT DE L'ESSENCE
4
AJOUTER LE COUT DU PEAGE
5
CALCULER LE TEMPS NECESSAIRE
6
CALCULER LE COUT DE L'ESSENCE
7
CALCULER LE TEMPS NECESSAIRE
8
IMPRIMER LE TEMPS DU TRAJET ET LE COUT
9
FIN

Ainsi : A=B — Signifie non une égalité au sens arithmétique mais une équivalence : les variables A et B ont même contenu.

A>B — Signifie que le contenu de A est plus grand que le contenu de B.

A><B — Signifie que le contenu de A est différent du contenu de B.

A<=B — Signifie que le contenu de A est plus petit ou égal au contenu de B.

A>=B — Signifie que le contenu de A est plus grand ou égal au contenu de B.

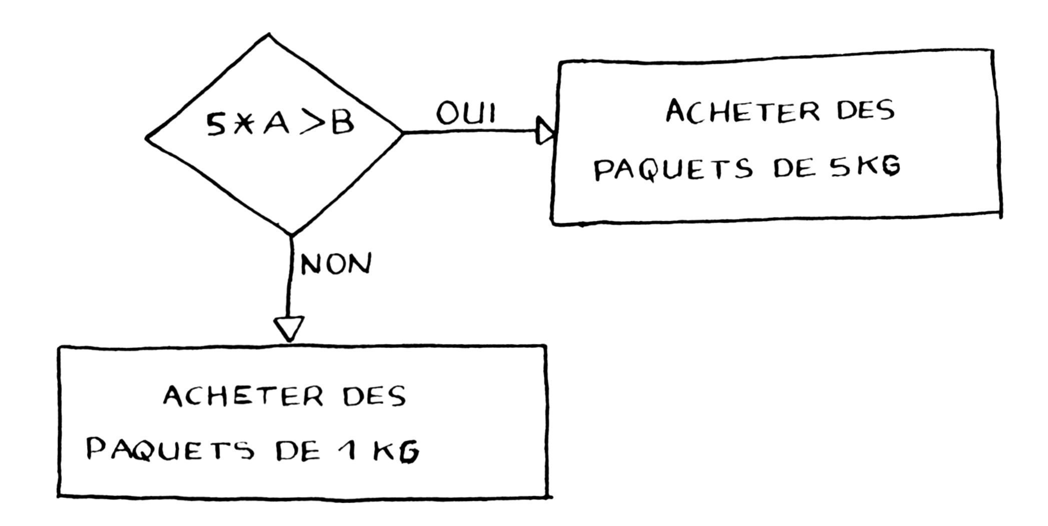
Dans quelle situation de la vie courante trouve-t-on ces conditions ? Faisons de nouveau appel à un exemple .

EXEMPLE IV . 2

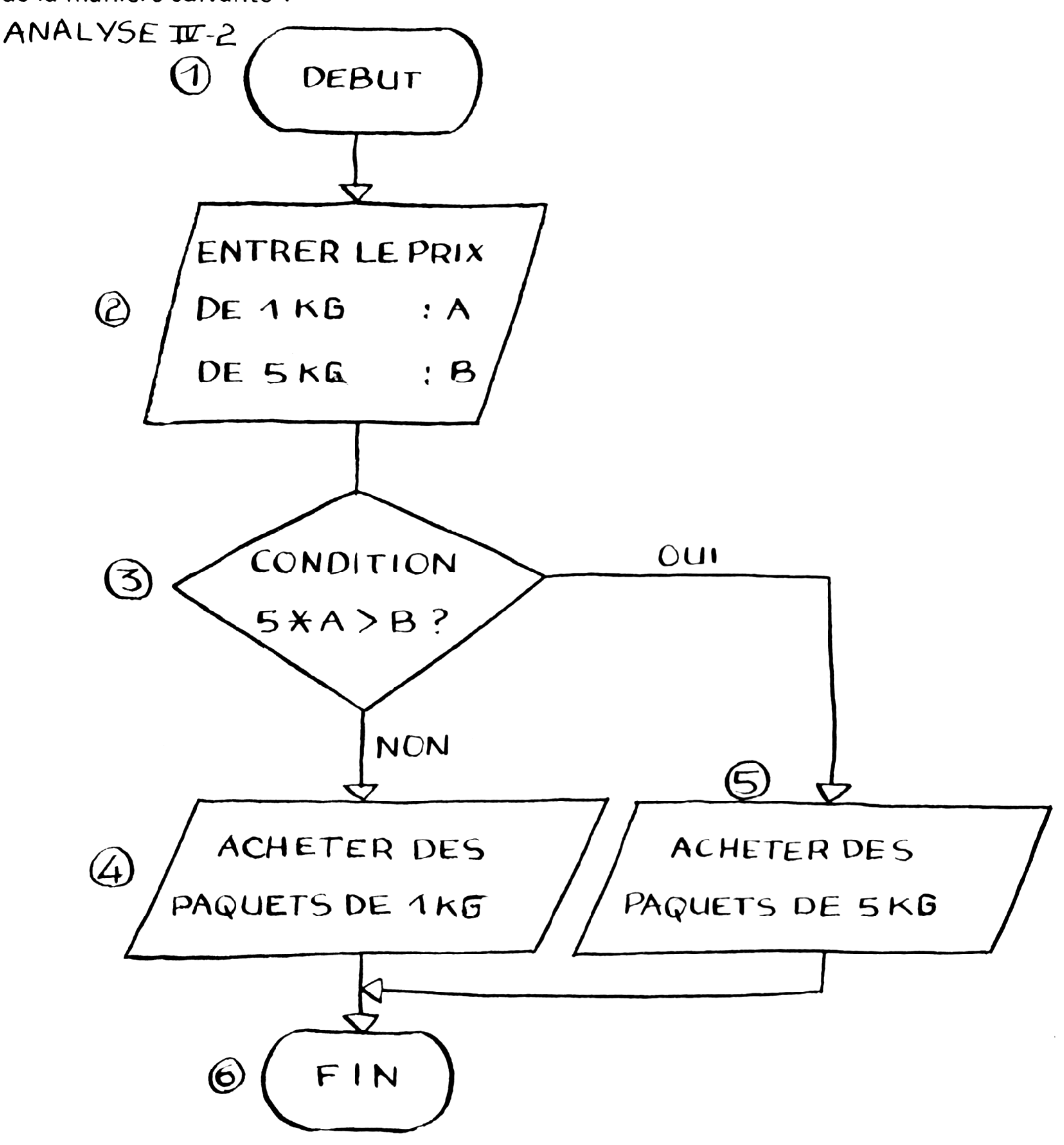
Dans un grand magasin, des paquets de lessive sont vendus sous deux conditionnements différents : le paquet de 1 kilo au prix de A Francs ; le paquet de 5 kilogrammes (Kg) au prix de B Francs.

Il est préférable d'acheter le paquet de 5 Kg si le prix de ce paquet est inférieur à cinq fois le prix du paquet de 1 Kg.

Le raisonnement peut se représenter ainsi :



Si ce problème simple devait être introduit sur ordinateur, il pourrait s'analyser de la manière suivante :



Remarquons que notre schéma présente à nouveau un branchement.

Le chemin OUI passe par (5) et (6)

Le chemin NON passe par (4) et (6)

Du point de vue de la programmation, il suffit de faire correspondre à chaque pavé les lignes de programmation telles que nous les avons vues dans le chapitre III. La condition sera exprimée par :

SI 5*A > B ALORS suivi du numéro de ligne correspondant à la case (5) .

En face des lignes de programmation, nous allons faire figurer les numéros de cases correspondants au schéma d'analyse.

PROGRAMME IV-2

```
(1)        NETTOIE
           NEW

(2)   5    ECRIS "PRIX DU PAQUET DE 1 KG "
           PRINT

     1Ø    DEMANDE A
           INPUT

     15    ECRIS "PRIX DU PAQUET DE 5 KG"
           PRINT

     2Ø    DEMANDE B
           INPUT

(3)  25    SI 5*A>B ALORS VATEN 4Ø
           IF         THEN  GOTO

(4)  3Ø    ECRIS "ACHETEZ DES PAQUETS DE 1 KG"
           PRINT

     35    VATEN 45
           GOTO

(5)  4Ø    ECRIS "ACHETEZ DES PAQUETS DE 5 KG"
           PRINT

(6)  45    FIN
           END
```

A l'usage ce programme fonctionnera, et nous devrons répondre aux interrogations de l'écran pour entrer les données A et B. Utilisons l'instruction EXE *(RUN)*.



PRIX DU PAQUET DE 1 KG

? 8

PRIX DU PAQUET DE 5 KG

? 35

ACHETEZ DES PAQUETS DE 5KG

8 et 35 sont les réponses que nous avons tapées sur le clavier, introduisant ainsi les valeurs A et B.

Pour reprendre un autre cas, il nous fait refaire EXE *(RUN)* et entrer de nouvelles données. Il est possible d'améliorer les performances de nos programmes. En effet, tous les programmes écrits jusqu'à présent nécessitent de revenir au début, de frapper EXE *(RUN)*, d'entrer à nouveau les données. On peut perfectionner le système en le faisant reprendre de lui-même le programme en cours ou au début de façon automatique. L'instruction VATEN peut être utilisée à cet effet. Nous verrons de quelle façon dans le chapitre suivant.

Celui-ci nous a permis de définir les termes d'*analyse* et d'utiliser l'instruction **SI.....ALORS** (*IF..... THEN*).

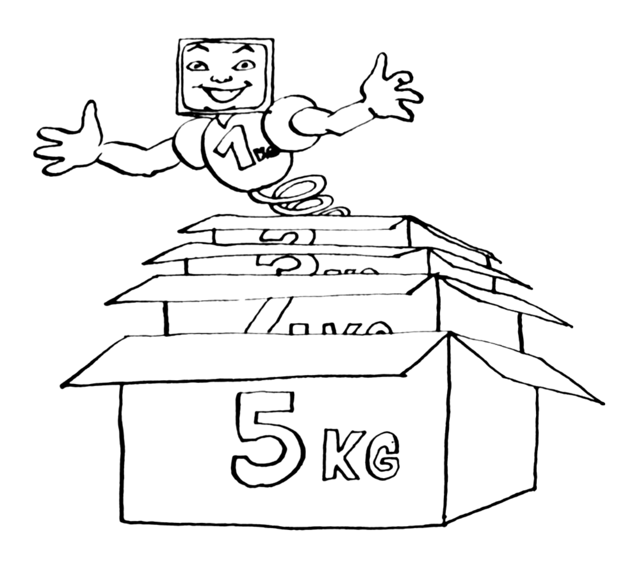
EXERCICE IV-1

Que se passe-t-il si dans le programme IV-2 nous supprimons la ligne 35 ?

EXERCICE IV-2

Modifier le programme IV-2 en posant la question :

5 * A <= B ?



EXERCICE IV-3

*Modifiez le programme de façon que si 5 * A=B, l'ordinateur affiche «FAITES CE QUE VOUS VOULEZ».*

EXERCICE IV-4

Vous avez remarqué que le programme IV-1 n'a pas été écrit. Faites-le.

BOUCLER, REPETER, COMPTER POUR MIEUX RAISONNER



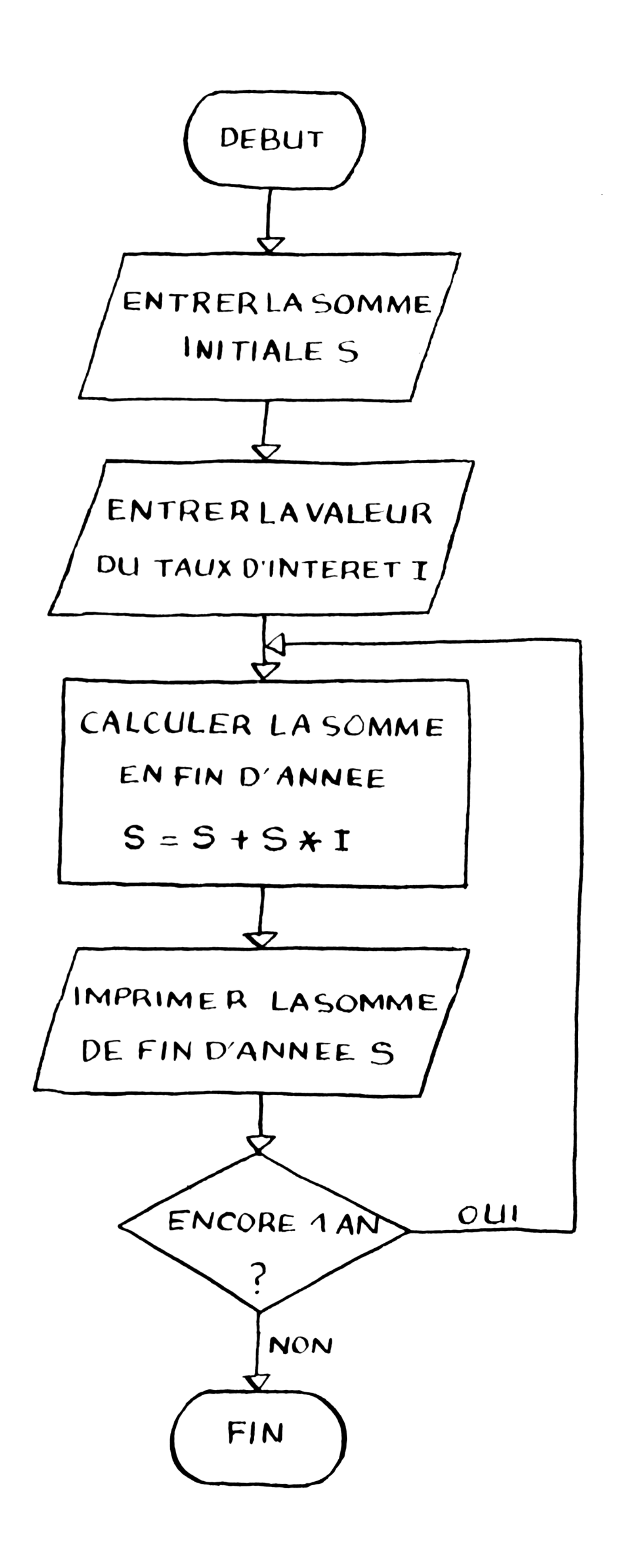
NOUS PLAÇONS DE L'ARGENT SUR UN COMPTE D'EPARGNE

Voyons maintenant comment de nouvelles instructions simplifient la tâche de l'utilisateur et lui évitent les redites, programmant l'ordinateur de façon à ce qu'il exécute les tâches répétitives.

La boucle de programmation

Revenons un instant sur l'instruction **VATEN** utilisée au chapitre III. Le programme utilisé correspondant à un calcul de consommation d'essence, appliquons cette même instruction au problème suivant.

Nous plaçons de l'argent sur un compte d'épargne au taux d'intérêt I. La somme initiale S devient égale à S+S * I à la fin de l'année. Une nouvelle année rapporte un nouvel intérêt calculé sur cette nouvelle somme et ainsi de suite ...


DEBUT
ENTRER LA SOMME INITIALE S
ENTRER LA VALEUR DU TAUX D'INTERET I
CALCULER LA SOMME EN FIN D'ANNEE
S = S + S * I
IMPRIMER LA SOMME DE FIN D'ANNEE S
ENCORE 1 AN ?
OUI
NON
FIN

Ce qui se concrétise par le programme suivant :

PROGRAMME V-1

```
5  ECRIS "QUELLE SOMME PLACEZ-VOUS ?"
   PRINT

1Ø DEMANDE S
   INPUT

15 ECRIS "A QUEL TAUX ANNUEL ?"
   PRINT

2Ø DEMANDE I
   INPUT

25 FAIS S=S+S*I
   LET

3Ø ECRIS "EN FIN D'ANNEES VOUS AVEZ"; S
   PRINT

35 ECRIS "LE PLACEZ-VOUS ENCORE UN AN ?"
   PRINT

4Ø DEMANDE R$
   INPUT

45 SI R$="OUI" ALORS VATEN 25
   IF          THEN  GOTO

5Ø FIN
   END
```

L'exécution de ce programme se fera de la façon suivante, à condition d'entrer la valeur du taux d'intérêt sous la forme exacte, à savoir 0.08 et non pas 8 %. Attention les ordinateurs, comme les Américains, utilisent le point à la place de la virgule.



QUELLE SOMME PLACEZ-VOUS ?

? 1∅∅∅

A QUEL TAUX ANNUEL

? ∅.∅8

EN FIN D'ANNEE VOUS AVEZ 1∅8∅

LE PLACEZ-VOUS ENCORE UN AN

?OUI

EN FIN D'ANNEE VOUS AVEZ 1166.4∅

LE PLACEZ-VOUS ENCORE UN AN

?

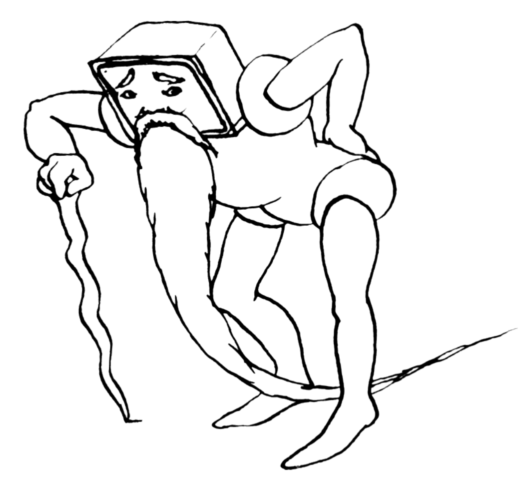
Le programme boucle sur lui-même, d'où le nom de *boucle* de programmation.

EXERCICE V-1

Le taux annuel I change chaque année. Quelle modification doit-on faire au programme pour en tenir compte ?

Le comptage

Le programme précédent ne permet pas de savoir le nombre d'années écoulées depuis la mise initiale. Nous devons nous rappeler ou noter le nombre de fois où nous avons répondu OUI. Confions cette tâche à la machine et désignons par N ce nombre d'années. Au début N=∅, puis, après 1 calcul N = 1
après 2 calculs N = 2 etc.



IL EST POSSIBLE D'IMPOSER A L'ORDINATEUR
D'AJOUTER +1 A CHAQUE PASSAGE

Il est possible d'imposer à l'ordinateur d'ajouter + 1 à chaque passage, ceci s'écrira :

FAIS N = N+1
LET

Rappelons-nous le chapitre III, relatif à l'instruction FAIS, où la variable N prend la valeur attribuée précédemment à N à laquelle on ajoute 1.

Ceci est vrai à condition qu'au début N ait une valeur égale à Ø. La mise à zéro de N s'appelle *initialisation de la variable* N, elle doit être faite avant d'entrer dans la boucle de programmation.

Le nouveau programme s'écrit alors :

PROGRAMME V-2

```
5   ECRIS "QUELLE SOMME PLACEZ-VOUS ?"
    PRINT

1Ø  DEMANDE S
    INPUT

15  ECRIS "A QUEL TAUX ANNUEL ?"
    PRINT

2Ø  DEMANDE I
    INPUT
```

```
23 FAIS N=Ø
   LET

25 FAIS S=S+S*I
   LET

27 FAIS N=N+1
   LET

30 ECRIS "APRES";N;"ANNEES VOUS AVEZ";S
   PRINT

35 ECRIS "LE PLACEZ-VOUS ENCORE UN AN?"
   PRINT

4Ø DEMANDE R$
   INPUT

45 SI R$="OUI" ALORS VATEN 25
   IF          THEN  GOTO

5Ø FIN
   END
```

Nous avons modifié le programme précédent par les instructions 23 et 27 qui effectuent le comptage. De plus l'instruction 3Ø permet de savoir où nous en sommes.

L'exécution de ce programme nous indiquera la somme S et le nombre d'années écoulées. A titre d'exercice, vous pouvez en imaginer le déroulement.

EXERCICE V-2

Que se passe-t-il si nous écrivons :

```
45 SI R$="OUI" ALORS VATEN 23
   IF          THEN  GOTO
```

Nous pouvons encore y apporter une amélioration. En effet, à priori nous désirons placer cette somme pour 5 ans et obtenir les sommes dont nous disposons à chaque fin d'année, sans avoir à introduire ces réponses OUI ou NON prévues dans le programme initial.

PROGRAMME V-3

Pour cela il suffit d'écrire (les instructions 5 à 30 sont identiques)

```
35 SI N<5 ALORS VATEN 25
   IF        THEN   GOTO

40 FIN
   END
```

et de supprimer les instructions 45 et 5∅.

Dans ce cas nous aurons successivement la valeur de la somme augmentée de ses intérêts en chaque fin d'année.

```
EXE
RUN
 |
 ▽
QUELLE SOMME PLACEZ-VOUS

?1∅∅∅

A QUEL TAUX ANNUEL

?0.08

APRES 1 ANNEES VOUS AVEZ 1.∅8∅

APRES 2 ANNEES VOUS AVEZ 1.166.4∅

APRES 3 ANNEES VOUS AVEZ 1.259.71

APRES 4 ANNEES VOUS AVEZ 1.36∅.49

APRES 5 ANNEES VOUS AVEZ 1.469.33
```

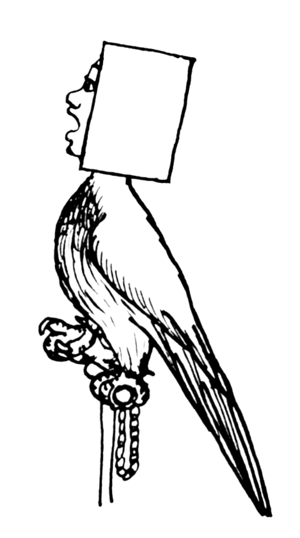
EXERCICE V-3

Remarquons au passage la faute de pluriel «1 ANNEES», mais que voulez-vous nous n'avons pas prévu le cas au singulier, il aurait fallu y penser plus tôt mais bien entendu cela est possible. FAITES-LE.

Pour aboutir au même résultat, il est possible d'utiliser une autre instruction. Ce sera la dernière que nous exploiterons dans ce livre.

INSTRUCTION :

REPETE ... ENCORE
FOR ... NEXT...



L'instruction : REPETE... ENCORE *(FOR... NEXT)*

Dans l'exemple précédent, nous savons ce que nous voulons : faire effectuer plusieurs fois la même partie de programme à notre machine.

C'est le rôle de l'instruction **REPETE** située au début de cette partie de programme et de **ENCORE** à la fin de cette partie de programme. (En Basic boucle FOR ... NEXT ...).

Pour résoudre notre problème sur 5 ans, modifions le programme V-3 de la façon suivante :

PROGRAMME V-4

```
5   ECRIS "QUELLE SOMME PLACEZ-VOUS ?
    PRINT

1Ø  DEMANDE S
    INPUT

15  ECRIS "A QUEL TAUX ANNUEL"
    PRINT

2Ø  DEMANDE I
    INPUT
```

```
23 REPETE N=1 JUSQUE 5
   FOR          TO

25 FAIS S=S+S*I
   LET

3Ø ECRIS "APRES";N;"ANNEES VOUS AVEZ";S
   PRINT

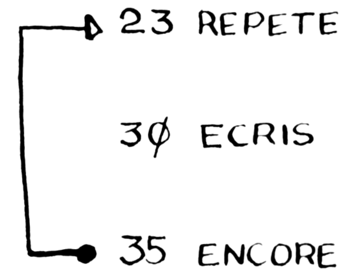
35 ENCORE N
   NEXT

4Ø FIN
   END
```

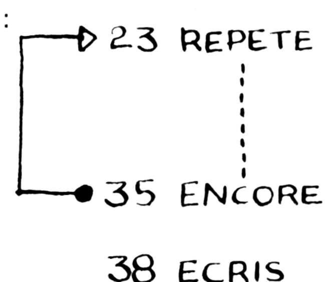
Comment le programme va-t-il se dérouler ? Lorsque la ligne 20 a été exécutée, on arrive à la ligne 23 qui indique que les lignes placées entre **REPETE** et **ENCORE** doivent être exécutées 5 fois. A chaque exécution, N est augmenté de 1 et la répétition s'arrête lorsque N atteint la valeur 5. (Bien entendu, en notant N = 2 jusque 6 donnerait le même résultat).

L'exécution de ce programme sera donc exactement identique à celle du précédent. Nous aurons une impression des résultats année par année.

En effet la structure du programme a placé l'instruction ECRIS *entre* les instructions **REPETE** et **ENCORE** ; il en résulte une écriture à chaque boucle :



Si la disposition du programme était telle que l'instruction ECRIS *soit en dehors* des instructions **REPETE** et **ENCORE**, nous n'aurions qu'une écriture pour N = 5, soit à la fin du programme :



38 ECRIS

Essayons maintenant de faire un programme qui ne porte plus sur 5 ans, mais sur un nombre d'années à préciser par l'opérateur.

La variable **J** désignera ce nombre.

Nous réécrivons le programme, en intercalant les lignes 21 et 22.

```
21 ECRIS "COMBIEN D'ANNEES"
   PRINT

22 DEMANDE J
   INPUT

23 REPETE N=1 JUSQUE J
   FOR        TO
```

L'instruction 23 est modifiée (J au lieu de 5)

Nous obtenons là un programme permettant de calculer la somme obtenue après **J** années. Le programme devient :

```
PROGRAMME V-5

5  ECRIS "QUELLE SOMME PLACEZ-VOUS ?"
   PRINT

1Ø DEMANDE S
   INPUT

15 ECRIS "A QUEL TAUX ANNUEL"
   PRINT

20 DEMANDE I
   INPUT

21 ECRIS "COMBIEN D'ANNEES"
   PRINT

22 DEMANDE J
   INPUT

23 REPETE N=1 JUSQUE J
   FOR        TO

25 FAIS S=S+S*I
   LET
```

```
35 ENCORE N
   NEXT

38 ECRIS "APRES"; N; "ANNEES VOUS AVEZ"; S
   PRINT

40 FIN
   END
```

NOUS OBTENONS LA UN PROGRAMME PERMETTANT DE CALCULER LA SOMME APRES J ANNEES



Remarquons au passage qu'il a été possible de fixer une limite au comptage en la désignant par une variable. Le comptage sera donc dépendant du contenu de cette variable, ici J.

L'instruction REPETE ... ENCORE est très puissante, l'exemple suivant de la table de multiplication par 7 est très significatif :

PROGRAMME V-6

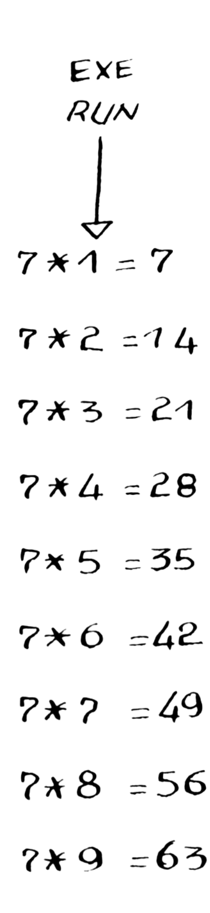
```
5  REPETE I=1 JUSQUE 9
   FOR        TO

10 FAIS A=I*7
   LET

15 ECRIS "7*"; I; "="; A
   PRINT

20 ENCORE I
   NEXT

30 FIN
   END
```



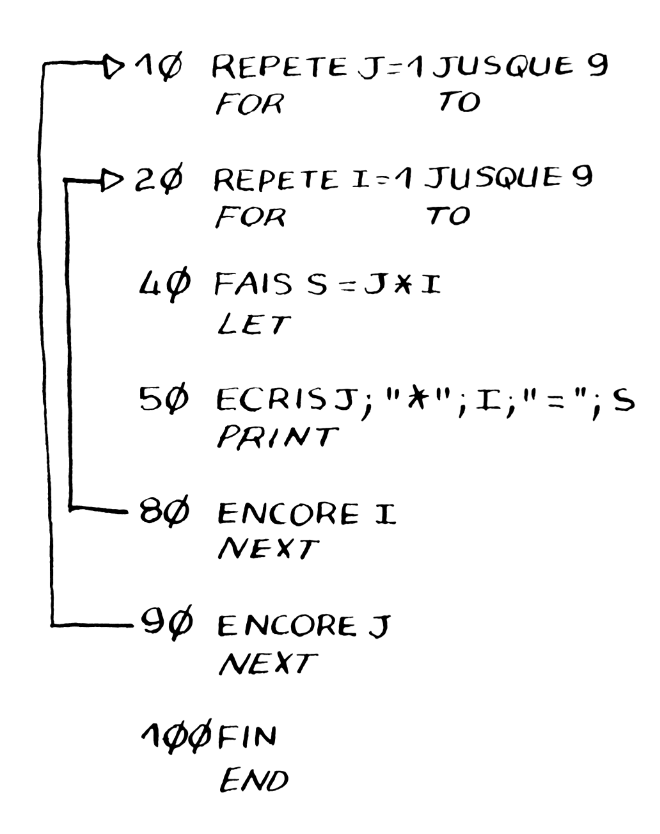
Cette fois l'ordinateur a effectué un travail important sans qu'il ait été besoin d'intervenir, montrant ainsi les possibilités qui sont offertes par cette instruction.

EXERCICE V-4

Que faut-il ajouter pour obtenir la somme des résultats ? (7+ 14+21 ..... +63).

Il est possible d'utiliser deux boucles de comptage, mais dans ce cas il faut prendre garde à ne pas les imbriquer l'une dans l'autre. Dans ce cas ENCORE doit être suivi obligatoirement de la désignation de la variable comme dans le programme V-7 ci-contre.

PROGRAMME V-7



```
10  REPETE J=1 JUSQUE 9
    FOR        TO

20  REPETE I=1 JUSQUE 9
    FOR        TO

40  FAIS S=J*I
    LET

50  ECRIS J;"*";I;"=";S
    PRINT

80  ENCORE I
    NEXT

90  ENCORE J
    NEXT

100 FIN
    END
```

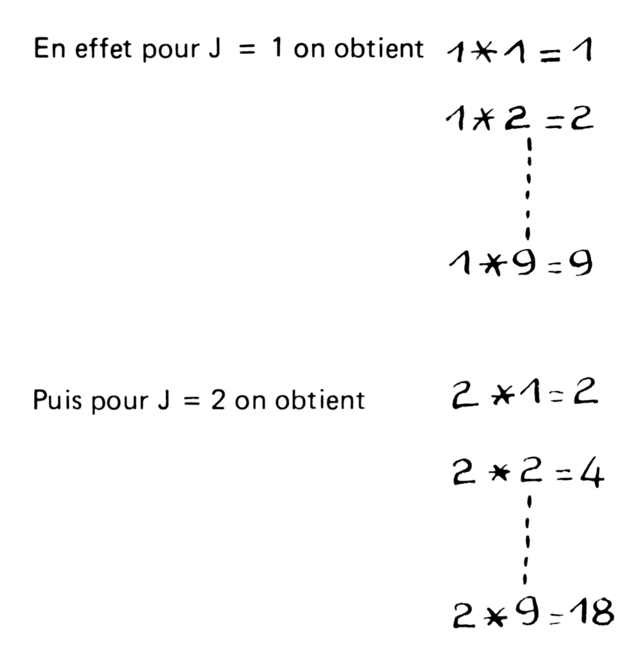
Ce programme permet d'obtenir toutes les tables de multiplications de 1 à 9 telles que nous les avons connues sur les bancs de l'école, au dos de nos cahiers.



CE PROGRAMME PERMET D'OBTENIR TOUTES LES TABLES DE MULTIPLICATION

En effet pour J = 1 on obtient

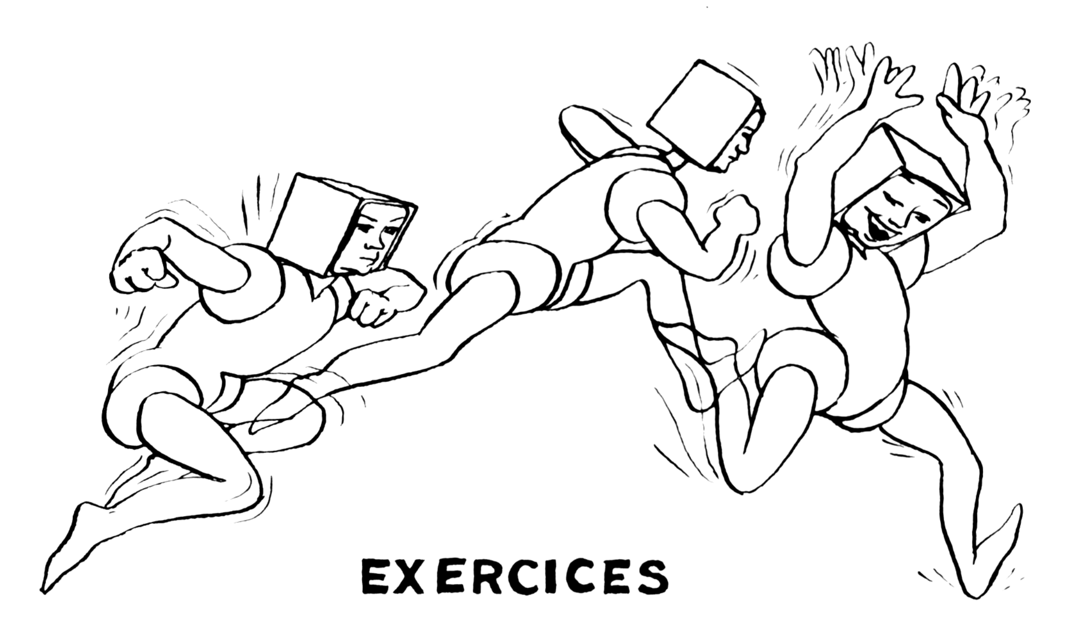
Puis pour J = 2 on obtient



et ainsi de suite jusqu'à 9.

Nous avons là des outils permettant de résoudre quelques petits problèmes. La **boucle**, le **comptage**, sont essentiels à la programmation Basic. Aller plus loin est possible.

Pour l'instant, contentons-nous de quelques exercices, qui par leur simplicité nous permettront de nous adapter au raisonnement informatique.



EXERCICES

Et maintenant, il nous reste à faire des exercices. En effet, la programmation, comme toute autre activité, nécessite de la pratique.

Nous vous proposons donc, au cours de ce chapitre, des exemples, et une solution de programmation.

Nous avons bien dit «**une**», car pour un même problème, il peut y avoir plusieurs chemins qui conduisent à une solution.

EXERCICE VI-1

Constituer un petit programme qui permette de tenir à jour sa comptabilité bancaire personnelle.

Tout d'abord nous remarquons qu'il est nécessaire de partir d'une somme initiale, d'en retrancher les chèques émis et d'y ajouter les chèques reçus. Il peut y avoir plusieurs opérations successives, il nous faut donc prévoir la possibilités de répéter émission ou retrait en plusieurs fois.

Lorsque toutes les opérations auront été passées, il faudra connaître la somme restante, figurant l'état du compte après ces opérations.

Cette analyse nous conduit au programme suivant :

```
5  ECRIS "INDIQUER L'ETAT DU COMPTE INITIALEMENT"
   PRINT

1Ø DEMANDE S
   INPUT

15 ECRIS "INDIQUER LE MONTANT DE L'OPERATION"
   PRINT

2Ø ECRIS "LE SIGNE -INDIQUE UNE DEPENSE"
   PRINT

25 ECRIS "LE SIGNE + INDIQUE UNE RENTREE"
   PRINT

3Ø DEMANDE M
   INPUT

35 FAIS S=S+M
   LET

4Ø ECRIS "VOULEZ-VOUS FAIRE UNE AUTRE OPERATION?"
   PRINT

45 ECRIS "Ø=OUI, 1=NON"
   PRINT

5Ø DEMANDE R
   INPUT

55 SI R=Ø ALORS 15
   IF      THEN

6Ø ECRIS "IL VOUS RESTE";S
   PRINT

65 FIN
   END
```

L'instruction 55 permet de recommencer autant de fois que cela s'avère nécessaire. Nous avons introduit en 50 une variable R qui correspond à la réponse atten-

due, avec deux possibilités : soit 0 à la place de OUI, soit 1 à la place de NON et dans ce cas le déroulement du programme est terminé.

EXERCICE VI-2

Etablir un programme qui permette d'entrer deux nombres : soit de les additionner, soit de les soustraire, soit de les multiplier, soit de les diviser l'un par l'autre.

L'analyse de cet exemple indique qu'il faut tout d'abord connaître ces deux nombres A et B, puis effectuer l'une des 4 opérations choisies.

Un tel programme peut s'écrire de la façon suivante :

```
5   ECRIS "DONNER DEUX NOMBRES A ET B"
    PRINT

1Ø  DEMANDE A, B
    INPUT

15  ECRIS "POUR A+B  FRAPPER 1"
    PRINT

2Ø  ECRIS "POUR A-B FRAPPER 2"
    PRINT

25  ECRIS "POUR A*B  FRAPPER 3"
    PRINT

3Ø  ECRIS "POUR A/B FRAPPER 4"
    PRINT

35  DEMANDE R
    INPUT

4Ø  SI R=1 ALORS 65
    IF     THEN

45  SI R=2 ALORS 75
    IF     THEN

5Ø  SI R=3 ALORS 85
    IF     THEN

55  SI R=4 ALORS 95
    IF     THEN
```

```
6Ø VATEN 15
   GOTO

65 FAIS C=A+B
   LET

7Ø VATEN 1ØØ
   GOTO

75 FAIS C=A-B
   LET

8Ø VATEN 1ØØ
   GOTO

85 FAIS C=A*B
   LET

9Ø VATEN 1ØØ
   GOTO

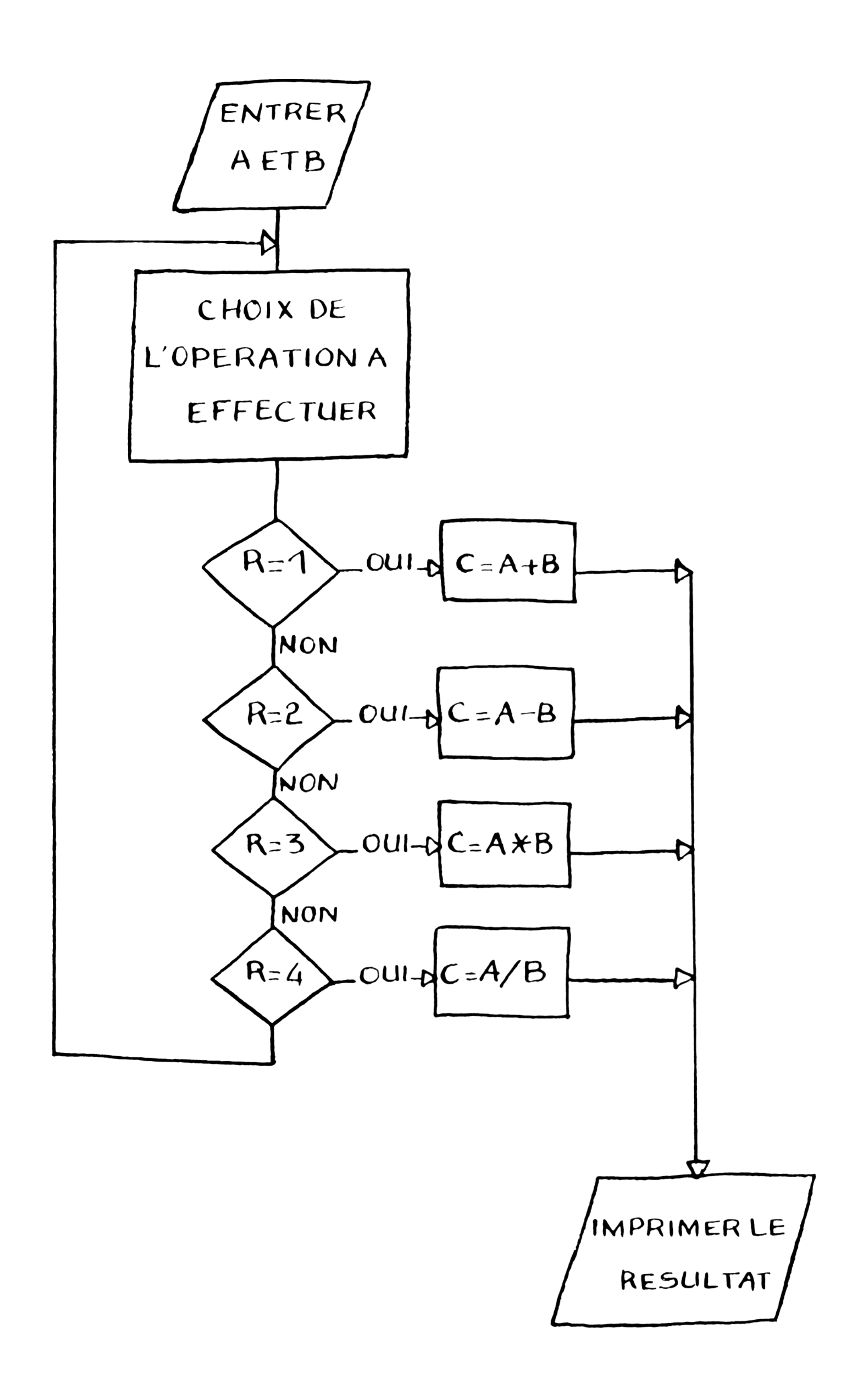
95 FAIS C=A/B
   LET

1ØØ ECRIS C
    PRINT
```

Nous avons introduit ici 4 possibilités de réponses pour R (de 40 à 55), qui se réduisent à 4 tests successifs.

Il est possible que l'opérateur se trompe et réponde autre chose qu'un chiffre allant de 1 à 4, nous avons prévu que dans ce cas l'ordinateur repose la question jusqu'à ce qu'une bonne réponse soit obtenue (instruction 60).

L'organigramme ci-contre donne l'architecture du programme ci-dessus.


ENTRER A ET B
CHOIX DE L'OPERATION A EFFECTUER
R=1
OUI
C=A+B
NON
R=2
OUI
C=A−B
NON
R=3
OUI
C=A*B
NON
R=4
OUI
C=A/B
IMPRIMER LE RESULTAT

EXERCICE VI-4

Deux personnes ont ensemble un certain nombre d'années N. Sachant que le premier a Y ans de plus que le second, quel est l'âge de chacun ?

Ce genre de problème peut s'appliquer à d'autres données (à des prix, à des quantités, des distances, des volumes). Il faut bien entendu commencer par connaître N et Y.

```
5   ECRIS "QUEL EST LEUR AGE TOTAL ?"
    PRINT

10  DEMANDE N
    INPUT

15  ECRIS "QUELLE EST LA DIFFERENCE D'AGE ENTRE"
    PRINT

20  ECRIS "LE PREMIER ET LE SECOND ?"
    PRINT

25  DEMANDE Y
    INPUT

30  FAIS B=(N-Y)/2
    LET

35  FAIS A=(N+Y)/2
    LET

40  ECRIS "LE PREMIER A"; A ;"ANS"
    PRINT

45  ECRIS "LE SECOND A"; B;"ANS"
    PRINT
```

Si A est l'âge du premier

Si B est l'âge du second

On a $A + B = N$

$A = B + Y$

où $B + B + Y = N$

et $B = \frac{N-Y}{2}$

Soit : $A = \frac{N-Y+2Y}{2} = \frac{N+Y}{2}$

Ce qui permet le programme (attention à la façon d'écrire la division, il n'y a pas de grande barre de fraction, mais le signe / qui doit ici porter sur une opération déjà effectuée, d'où l'utilisation des parenthèses).

```
30  FAIS B=(N-Y)/2
    LET

35  FAIS A=(N+Y)/2
    LET

40  ECRIS "LE PREMIER A"; A ;"  ANS"
    PRINT

45  ECRIS "LE SECOND A"; B ;"  ANS"
    PRINT
```

EXERCICE VI-5

ATTENTION : cet exemple ne peut être réalisé tel quel sur tous les systèmes, à cause du nombre de variables possibles. Il peut nécessiter quelques aménagements portant sur la désignations de ces variables.

Editer une facture comportant :

le nom, l'adresse du client ;

la quantité de chaque produit ;

le prix unitaire ;

la somme à payer ;

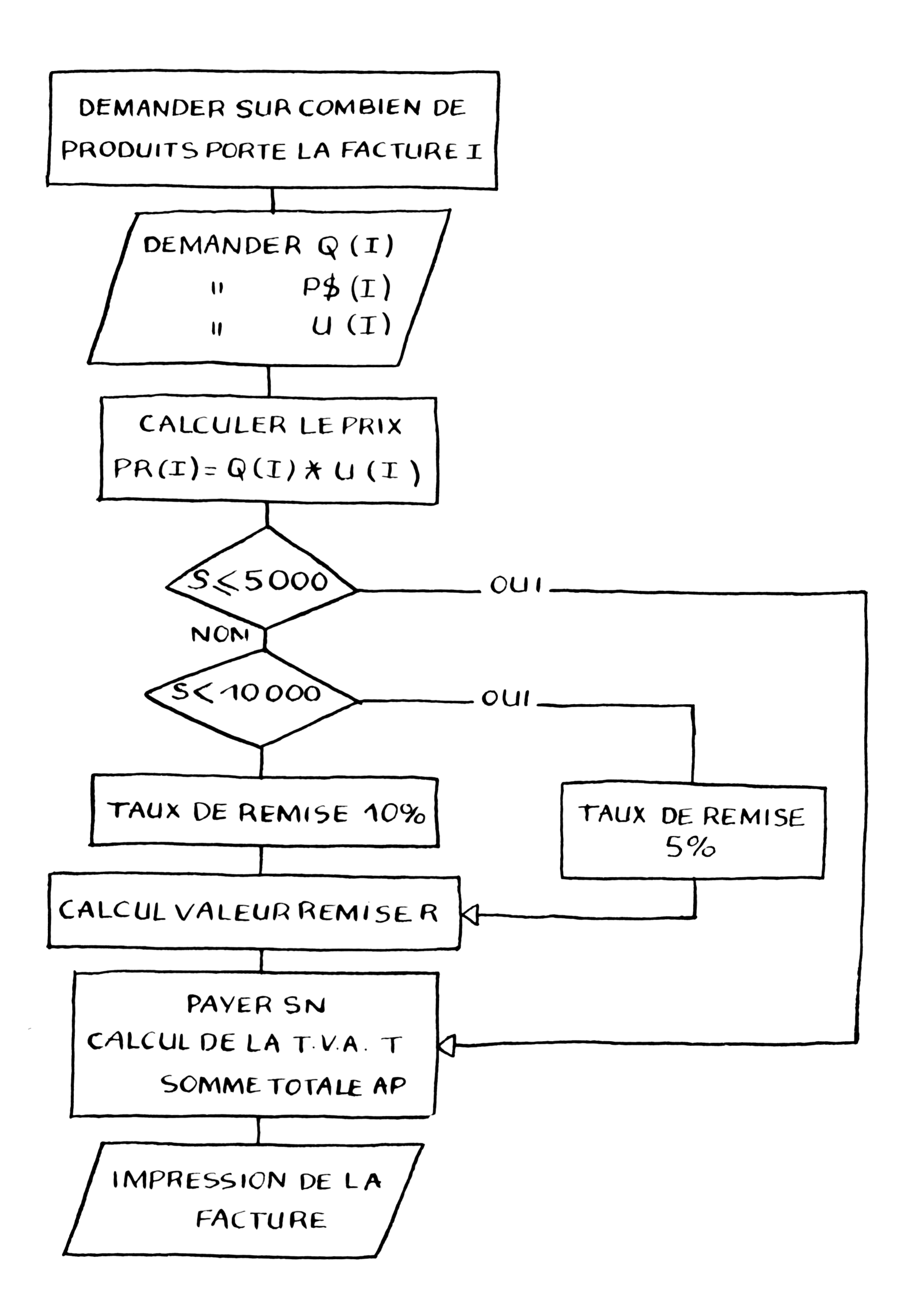
la remise égale à 5 % si la somme à payer est supérieure à 5 000 F. ;

la remise égale à 10 % si la somme à payer est supérieure à 10 000 F.

Nous allons d'abord indiquer à l'ordinateur les quantités **Q (I)** pour le produit **P$ (I)** au prix unitaire **U (I)**.

I peut prendre plusieurs valeurs 1, 2, 3, ... Nous devrons au préalable signaler à l'ordinateur sur combien de produits porte la facture.

L'écriture **Q (I)** est une façon nouvelle pour nous de noter une variable. **P$ (I)** signifie que nous pouvons écrire le nom du produit puisque la variable contient le symbole **$**.


DEMANDER SUR COMBIEN DE PRODUITS PORTE LA FACTURE I
DEMANDER Q (I)
" P$ (I)
" U (I)
CALCULER LE PRIX
PR(I) = Q(I) * U (I)
S ≤ 5000
OUI
NON
S < 10 000
OUI
TAUX DE REMISE 10%
TAUX DE REMISE 5%
CALCUL VALEUR REMISE R
PAYER SN
CALCUL DE LA T.V.A. T
SOMME TOTALE AP
IMPRESSION DE LA FACTURE

```
5    ECRIS "NOM ET ADRESSE DU CLIENT"
     PRINT

1Ø   DEMANDE A$
     INPUT

15   ECRIS "COMBIEN DE PRODUITS DIFFERENTS ?"
     PRINT

2Ø   DEMANDE B
     INPUT

25   ECRIS "INDIQUER DANS L'ORDRE LA QUANTITE, LE NOM, LE PRIX
     PRINT

     UNITAIRE"

3Ø   REPETE I=1 JUSQUE B
     FOR        TO

35   DEMANDE Q(I), P$(I), U(I)
     INPUT

4Ø   FAIS PR(I)=Q(I)*U(I)
     LET

45   FAIS S=S+PR(I)
     LET

5Ø   ENCORE I
     NEXT

55   SI S<=5.ØØØ ALORS VATEN 14Ø
     IF          THEN  GOTO

6Ø   SI S<1Ø.ØØØ ALORS VATEN 1ØØ
     IF          THEN  GOTO

65   FAIS R=S*1Ø/1ØØ
     LET
```

```
70  VATEN 145
    GOTO

100 FAIS R=S * 5/100
    LET

110 VATEN 145
    GOTO

140 FAIS R=0
    LET

145 FAIS SN=S-R
    LET

150 FAIS T=SN * 17,6/100
    LET

155 FAIS AP=SN+T
    LET

160 ECRIS A$
    PRINT

165 ECRIS "QUANTITE", "NOM", "PRIX UNITAIRE"
    PRINT

170 REPETE I=1 JUSQUE B
    FOR            TO

175 ECRIS Q(I), P$(I), U(I)
    PRINT

180 ENCORE I
    NEXT

185 ECRIS "VALEUR DUE";S
    PRINT
```

```
190 SI R=0 ALORS VATEN 200
    IF       THEN  GOTO

195 ECRIS "REMISE"; R
    PRINT

200 ECRIS "TAXE"; T
    PRINT

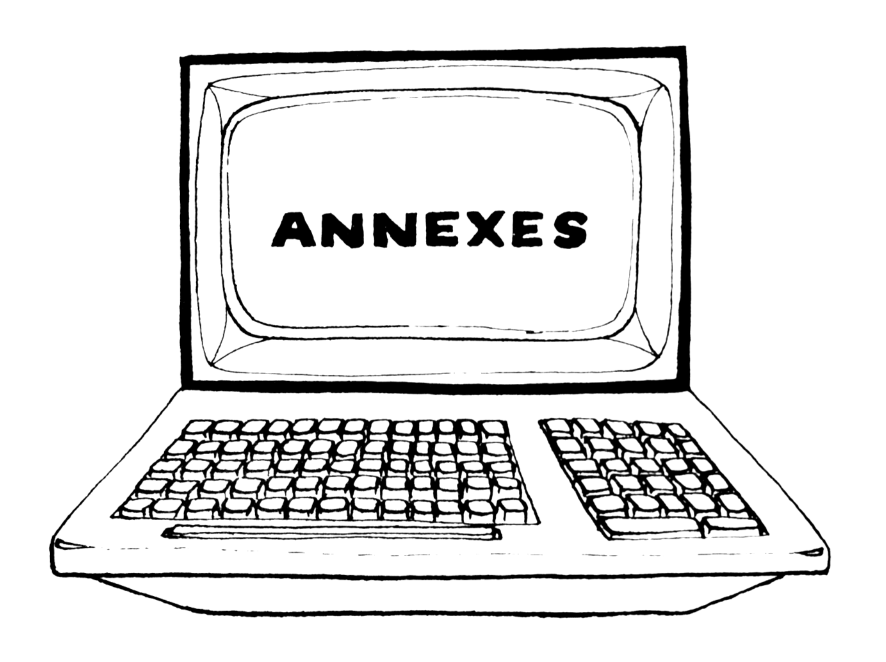
210 ECRIS "TOTAL A PAYER"; AP
    PRINT
```

Par cet exercice nous voici parvenus à réaliser une partie du travail de notre employé qui consiste en l'édition d'une facture.

Vous qui avez un ordinateur à votre disposition, vous pouvez en améliorer la présentation, vous connaissez maintenant les instructions nécessaires.

Vous pouvez aussi vouloir automatiser d'autres fonctions, pour cela il vous faut apprendre d'autres mots du vocabulaire Basicois ou Basic, il vous faut poursuivre l'aventure informatique.

Si nous vous en avons donné le goût, nous aurons atteint notre objectif ; vous pouvez maintenant passer à d'autres ouvrages tout aussi sérieux que celui-ci même si leur abord paraît plus austère.


ANNEXES

ANNEXE I
SOLUTION DES EXERCICES

EXERCICE III-1

```
10  ECRIS "NOM"
    PRINT

15  ECRIS "PRENOM"
    PRINT
```

En effet, chaque ligne de programme est exécutée à son tour, et sur l'écran y correspond une ligne d'écriture.

EXERCICE III-2

```
NOM, PRENOM
```

La différence avec l'exemple du texte vient de ce que nous n'avons pas utilisé la , en dehors des guillemets, et *tout* se qui se trouve entre elles est pris en compte.

EXERCICE III-3

```
15  DEMANDE A$
    INPUT

25  DEMANDE B$
    INPUT

30  ECRIS A$, B$
    PRINT
```

Nous avons utilisé B$ car le numéro d'immatriculation comporte des lettres, il faut y penser.

EXERCICE III-4

Lors des instructions ECRIS, il faut tenir compte de la composition du tableau ce qui donne (les espaces vides entre guillemets sont pris en compte) :

```
30   ECRIS "CONSOMMATION  DISTANCE  CONSOMMATION"
     PRINT

35   ECRIS "AUX 100 KM   PARCOURUE      TOTALE   "
     PRINT

40   ECRIS A;"      ";B;"    ";C
     PRINT
```

De façon pratique le nombre d'espaces blancs à positionner dans les guillemets est à compter de façon précise, tout comme le fait une dactylo lorsqu'elle compose un tableau.

Sur un ordinateur de poche, il n'est pas possible de faire apparaître un tel tableau puisqu'il n'y a qu'une ligne qui apparaît sur l'écran.

EXERCICE III-5

Dans le premier cas le résultat de A n'*apparaît pas sur l'écran, par* contre on nous redemande la distance parcourue de façon répétitive.

Dans le second cas le résultat de A *n'apparaît pas sur l'écran,* et nous sommes conduits à la fin sans nouvelle interrogation.

EXERCICE IV-1

Dans ce cas lorsque 5*A est < = B (plus petit ou égal à B) la réponse de la machine sera :

```
ACHETEZ DES PAQUETS DE 1 KG
ACHETEZ DES PAQUETS DE 5 KG
```

Ce qui laisse l'opérateur perplexe ...

EXERCICE IV-2

Voici une solution, vous pouvez en avoir une autre :

```
25  SI 5*A   = B ALORS VATEN 30
    IF             THEN  GOTO
```

```
27   VATEN 4Ø
     GOTO
```

Tout le reste est inchangé.

EXERCICE IV-3

Voici une solution :

```
25   SI 5*A    B ALORS VATEN 40
     IF          THEN   GOTO

27   SI 5*A    B ALORS VATEN 42
     IF          THEN   GOTO

3Ø   ECRIS "ACHETEZ DES PAQUETS DE 1 KG"
     PRINT

35   VATEN 45
     GOTO

4Ø   ECRIS "ACHETEZ DES PAQUETS DE 5 KG"
     PRINT

41   VATEN 45
     GOTO

42   ECRIS "FAITES CE QUE VOUS VOULEZ"
     PRINT

45   FIN
     END
```

EXERCICE IV-4

Une solution.

Voici la liste des variables utilisées :

P = Prix du carburant

A = Distance par autoroute

R = Distance par route nationale

V = Vitesse par autoroute

B = Consommation par autoroute aux 100 Km
W = Vitesse par route
D = Consommation sur route aux 100 Km
E = Péage sur autoroute.

Remarquons qu'il y a des variables indépendantes du véhicule, il s'agit de A et R. Nous pouvons les entrer une fois pour toutes dans le programme.

Par contre P, prix du carburant peut varier non seulement avec le temps (hélas), mais aussi avec le lieu où l'on fait le plein, et enfin la qualité du carburant (essence ordinaire, super, gasole).

Le prix du péage peut dépendre de la cylindrée du véhicule.

En sortie nous désirons obtenir le coût et le temps. Ces variables seront appelées

T pour le temps
C pour le coût

```
5    FAIS A = 25Ø
     LET

1Ø   FAIS R = 3ØØ
     LET

2Ø   DEMANDE V,B
     INPUT

25   ECRIS "VITESSE, CONSOMMATION SUR ROUTE"
     PRINT

3Ø   DEMANDE W,D
     INPUT

35   ECRIS "PRIX DU CARBURANT"
     PRINT

4Ø   DEMANDE P
     INPUT

45   ECRIS "PRIX DU PEAGE"
     PRINT

5Ø   DEMANDE E
     INPUT
```

```
55  ECRIS "ROUTE = Ø, AUTOROUTE = 1"
    PRINT

6Ø  DEMANDE R
    INPUT

65  SI R = Ø ALORS VATEN 1ØØ
    IF       THEN   GOTO

7Ø  FAIS C = (P*A*B/1ØØ) + E
    LET

75  FAIS T = A/V
    LET

8Ø  VATEN 12Ø
    GOTO

1ØØ FAIS C = P*R*D/1ØØ
    LET

1Ø5 FAIS T = R/W
    LET

12Ø ECRIS "TEMPS";T
    PRINT

125 ECRIS "COUT";C
    PRINT

13Ø VATEN . . . .
    GOTO
```

Nous vous laissons compléter l'instruction 130 ; le choix du VATEN *(GOTO)* peut vous permettre de repartir d'où vous souhaitez dans le programme.

EXERCICE V-1

Nous devons chaque année introduire I. Il suffit d'écrire :

```
45  SI RØ = "OUI" ALORS VATEN 15
    IF              THEN   GOTO
```

EXERCICE V-2

Dans ce cas à chaque boucle N est remis à 0, il n'y a pas de comptage.

EXERCICE V-3

Il nous faut prévoir à l'impression le cas où N = 1.

Le début du programme reste identique. Nous le modifions de la façon suivante :

```
30  SI N = 1 ALORS VATEN 40
    IF         THEN  GOTO

35  ECRIS "APRES";N;"ANNEES VOUS AVEZ";S
    PRINT

37  VATEN 45
    GOTO

40  ECRIS "APRES 1 ANNEE VOUS AVEZ";S
    PRINT

45  ECRIS "LE PLACEZ-VOUS ENCORE UN AN"
    PRINT

50  DEMANDE R$
    INPUT

55  SI R$ = "OUI' ALORS VATEN 25
    IF              THEN  GOTO

60  FIN
    END
```

EXERCICE V-4

Le programme V-6 est modifié en ajoutant :

```
7   FAIS S = 0
    LET

17  FAIS S = S+A
    LET
```

```
25  ECRIS "TOTAL=";S
    PRINT
```

L'instruction 7 initialise la somme à 0.

L'instruction 7 initialise la somme à 0.

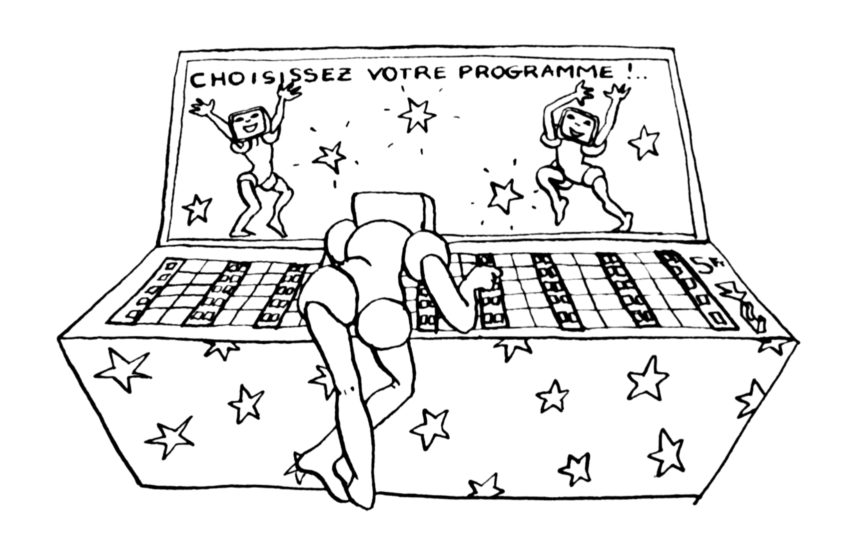
L'instruction 17 effectue la somme à chaque passage, c'est pourquoi elle est placée *au cœur d*e la boucle.

L'instruction 25 donne le résultat global, elle est donc *en dehors* de la boucle.

ANNEXE II
LEXIQUE BASICOIS ~ BASIC

CLASSES SUR LE MOT BASICOIS			CLASSES SUR LE MOT BASIC		
		Page d'apparition			Page d'apparition
ALORS	*THEN*	44	*END*	FIN	33
ARRET	*STOP*	39	*FOR*	REPETE	58
DEMANDE	*INPUT*	28	*GOTO*	VATEN	37
ECRIS	*PRINT*	27	*IF*	SI	44
ENCORE	*NEXT*	58	*INPUT*	DEMANDE	28
EXE	*RUN*	27	*LET*	FAIS	34
FAIS	*LET*	34	*LIST*	LISTE	26
FIN	*END*	33	*NEW*	NETTOIE	33
JUSQUE	*TO*	59	*NEXT*	ENCORE	58
LISTE	*LIST*	26	*PRINT*	ECRIS	27
NETTOIE	*NEW*	33	*RUN*	EXE	27
REPETE	*FOR*	58	*STOP*	ARRET	39
SI	*IF*	44	*THEN*	ALORS	44
VATEN	*GOTO*	37	*TO*	JUSQUE	59

ANNEXE III
PETIT GLOSSAIRE INDEXÉ


CHOISISSEZ VOTRE PROGRAMME !..

Achevé d'imprimer en mai 1981
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : 2e trimestre 1981

N° d'impression : 20061
N° d'édition : 86595-21-1

Composition Liliane COLLARD

L'informatique et les ordinateurs ont déjà fait couler beaucoup d'encre; l'auteur a décidé d'en faire couler un peu plus, mais très peu, pour tenter de les faire découvrir aux "ignorants".
Jean-Michel JEGO a délibérément décidé de vous séduire, de vous prendre par la manière douce (pas de calcul binaire, pas d'algèbre de Boole...), de vous présenter les bons côtés de l'informatique individuelle.
Le "visa" dont il est question est donc valable pour une rapide et confortable excursion et non pour un long et aride voyage.
Une fois ce premier voyage terminé, vous pourrez songer à de nombreux autres, ou vivre sur vos souvenirs, en tout état de cause, vous aurez passé un bon moment.



Jean-Michel Jégo est docteur ès-Sciences. Après avoir réalisé des travaux de recherche scientifique, il a animé des actions de formation continue en entreprises. Cette expérience lui a permis de juger de l'importance à accorder aux entrées en matière, à la qualité de l'initiation.



Editions du P.S.I. - ISBN 2-86595-021-2
41-51, rue Jacquard 77400 Lagny (France)
Tél. (6) 007.59.31

Editions du P.S.I.

VISA POUR L'INFORMATIQUE

Jean-Michel Jégo

Document numérisé avec amour par :



AMSTRAD CPC

MÉMOIRE ÉCRITE

ACME

https://acpc.me/